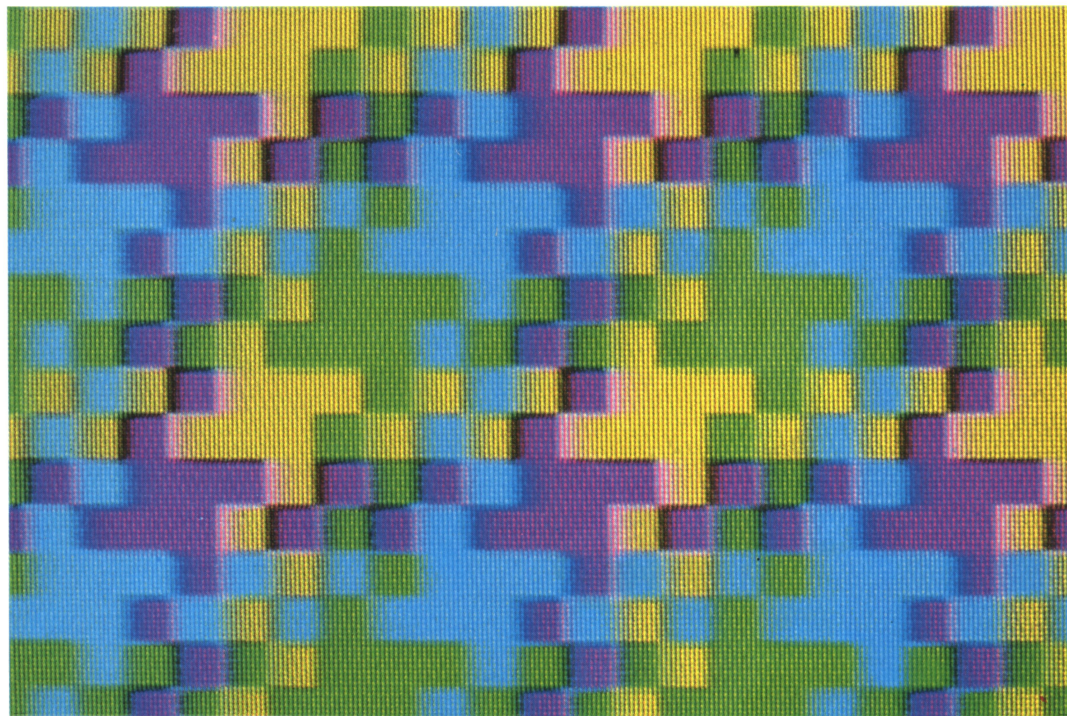




Dal libro di testo all'uso dello Spectrum



Risorse per la Didattica



Dal libro di testo all'uso dello Spectrum

Risorse per la Didattica



GRUPPO
EDITORIALE
JACKSON
Via Rosellini, 12
20124 Milano

© Copyright Gruppo Editoriale Jackson - Novembre 1985

GRAFICA E IMPAGINAZIONE: Confalone Moreno

COPERTINA: Silvana Corbelli

FOTOCOMPOSIZIONE: Jo Type - Pero, Milano

STAMPA: Grafika 78 - Pioltello - Milano

Tutti i diritti sono riservati. Stampato in Italia. Nessuna parte di questo libro può essere riprodotta, memorizzata in sistemi di archivio, o trasmessa in qualsiasi forma o mezzo, elettronico, meccanico, fotocopia, registrazione o altri senza la preventiva autorizzazione scritta dell'editore.

PRESENTAZIONE

Il lavoro che qui si presenta è il risultato del primo anno di lavoro sul progetto "Banca di Unità Didattiche Software" proposto e realizzato da Risorse per la didattica s.r.l.

Il progetto fu messo in cantiere dopo un affollato seminario ai margini della mostra, allestita alla fiera di Milano dalla Provincia, sul libro scientifico e tecnico.

In quella sede sembrava di poter convenire con le centinaia di colleghi presenti, su alcuni punti da verificare in concreto.

Il primo punto era che nella scuola di base dovesse essere tentato sia l'insegnamento delle varie discipline con l'ausilio del computer, sia l'insegnamento di più linguaggi per programmare il computer.

Il secondo punto era che una macchina piccola e poco costosa fosse preferibile ad una macchina grande e sofisticata se possedeva almeno due linguaggi interprete ed una possibile espansione perché prometteva una diffusione nelle scuole con rapidità rapportabile all'obsolescenza.

Il terzo punto era che il software didattico non dovesse essere accettato in assoluto come una "scatola chiusa" ma potesse essere letto, interpretato, trasformato e utilizzato flessibilmente dagli stessi docenti così come è possibile leggere, interpretare, utilizzare flessibilmente un testo scolastico.

Il quarto punto era che fosse necessario, per conseguenza, sostenere il software con un supporto scritto che fosse insieme il libro delle istruzioni per l'uso e degli iter didattici possibili.

Si è formata un'equipe multidisciplinare che ha proposto una strategia e degli esempi: la strategia è quella di lavorare su più livelli di complessità crescente.

Il primo livello ha preso il nome di U.D.S. Per ora il livello più complesso è formato da piccoli percorsi didattici, ma si intravede la possibilità di comporre e correlare più percorsi in modo flessibile per realizzare dei veri corsi di studio.

Per ogni livello vengono fatti degli esempi e vengono presentati con il corredo scritto secondo le indicazioni prima esplicitate: istruzioni per l'uso e iter didattici possibili.

Nel primo capitolo abbiamo formulato le linee di una strategia per la diffusione dell'informatica nella scuola di base.

Il titolo del capitolo, "Per una strategia", ha carattere non definitivo.

Si vuole sottintendere che questo nostro primo intervento, è una delle voci possibili da raccogliere per formulare una strategia alla quale siamo convinti di poter dare un valido contributo.

Risorse per la didattica

L'équipe del progetto "Banca di Unità didattiche software" è così composta:

PIERO D'ALFONSO,
GIOVANNI MARTORANA

Responsabili del progetto

FRANCO DE ANNA

Esperto di politica scolastica

TERESA DEL MASCHIO
LUISA TINELLI
SERGIO VETTORE

Esperti di matematica

LUIGI D'ANGELO
LUCIA LEOMBRUNO
LIVIO MANDELLI

Esperti di educazione linguistica

FRANCO STILLO

Esperto di educazione tecnica

L'équipe ringrazia i professori Alba Rossi Dell'Acqua, Giulio Mezzetti, Riccardo Luccio per la preziosa consulenza prestata.

INDICE

- CAPITOLO I: PER UNA STRATEGIA** 1
 - 1.1. Formazione, istruzione e uso del computer 1
 - 1.2. Unità didattiche software 17
- CAPITOLO II: ESEMPI U.D.S.** 23
 - 2.1. Scomposizione in fattori 23
 - 2.2. Da base 2 a base 10 30
 - 2.3. Punti sul piano cartesiano 37
 - 2.4. Proiezione di punti su assi cartesiani 44
 - 2.5. Traslazione di un punto sul piano 48
 - 2.6. Disegno di figure sul piano cartesiano 54
 - 2.7. Traslazione di figura sul piano cartesiano 62
- CAPITOLO III: ESEMPI DI SOFTWARE DIDATTICO** 73
 - 3.1. La rappresentazione di tessuti: una simulazione per l'educazione tecnica 73
 - 3.2. Istogrammi: una utility per la geografia e le scienze 105
- CAPITOLO IV: PERCORSI DIDATTICI** 149
 - 4.1. La statistica dei grandi numeri spiegata con carta, matita e con il computer: una simulazione 149
 - 4.2. I quadrilateri 182
 - 4.3. Educazione linguistica: i Giudizi del Parlante 224

Per una strategia

1.1. Formazione, istruzione e uso del computer

FRANCO DE ANNA

Parlare di scuola significa parlare di uno dei sistemi fondamentali della struttura sociale e più precisamente di quello che si occupa, per dovere istituzionale, della trasmissione culturale.

C'è una necessità tassativa nel "modello umano di sopravvivenza" ed è quella di non lasciare che si disperda, nel passaggio da una generazione all'altra, il patrimonio di conoscenze ed abilità acquisto nel passato dato che esso si arricchisce in modo estremamente rapido e non ci si può affidare alla sola trasmissione genetica.

Non è qui la sede adatta per articolare questa riflessione, ma si ritiene utile ricorrervi per poter rimandare alle ragioni da cui muovono due filoni caratteristici del nostro sistema scolastico: quello che si occupa della formazione della singola persona, e quello che riguarda la sua istruzione.

Sostanzialmente fare scuola significa due cose:

la prima rendere l'uomo agile di fronte agli ostacoli che dovrà superare, sviluppando la sua capacità di scegliere, la rapidità di decidere e di giudicare, la disposizione a prefigurare il risultato di una strategia;

la seconda dotarlo di conoscenze, abilità e informazioni sufficienti ad elaborare in concreto le strategie, le procedure, le operazioni necessarie a sopravvivere, ad adattare l'ambiente a sé, a fornire il proprio personale contributo ai bisogni collettivi.

È evidente che i due aspetti sono fortemente correlati fra loro.

Come è possibile sviluppare una sia pur minima capacità di scegliere in assenza di conoscere e abilità elementari? E, al contrario, la conoscenza di quali operazioni o procedure può essere trasmessa e appresa da individui che non siano in grado di valutarne l'efficacia o di controllarne le fasi?

La forte correlazione tra formazione dell'individuo e istruzione propria, è tale da configurare un feed back che sarà tanto più positivo quanto maggiore sarà stata la concatenazione tra le valenze formative e istruttive nel suo corso di studi.

Se è la madre che fa da innesco iniziale in questo processo, è la scuola che si propone in seguito come l'ambiente del suo sviluppo.

Fare bene scuola vuol dire, perciò, curare le implicazioni formative, quelle istruttive e la loro concatenazione per ottenere il massimo dell'efficacia, ma vuol dire anche riconoscere la particolare "forma" del proprio "ambiente" che non è riconducibile solo ai confini di una certa struttura muraria, ma si deve considerare espanso a tutti quei luoghi quali la palestra di pallavolo, la sala cinematografica, la biblioteca, il teatro, l'ospedale e a tutte quelle situazioni, come la lettura di un testo, la gestione di un elaboratore, l'uso di una video cassetta, l'ascolto di un disco, la lettura di una porzione di cielo, l'analisi di un miscuglio al microscopio, lo scatto di una fotografia dove si metta in atto una concatenazione tra valenze formative e valenze istruttive.

Secondo questo principio fa cattiva scuola chi trascura una di queste due valenze e non fissa strategie adeguate per tenerle tra loro correlate.

Non fa però buona scuola neppure colui che, pur stabilendo un corretto rapporto tra formazione e istruzione all'interno delle proprie "mura", non resta aperto ai luoghi e alle situazioni in cui possono sorgere nuove istanze di formazione o di situazioni e correlazioni tra le due valenze.

Può avvenire infatti, e in una notevole misura sta avvenendo, che momenti di grande e rapida trasformazione sociale come il nostro, determinino bisogni di formazione e di istruzione cui siano in grado di rispondere ambienti esterni alle mura scolastiche.

In questo caso, o questi ambienti sono così esperti di sistemi pedagogici da saper istituire la giusta correlazione tra valenze formative e valenze istruttive, e allora la risposta al bisogno sarà ben finalizzata, oppure questi ambienti non sono esperti e allora ne risulterà una risposta sfuocata.

Il rischio di un uso poco attento a tutti i bisogni formativi di un insegnamento esterno al sistema scolastico è dunque elevato così come è elevato il rischio di un insegnamento interno al sistema scolastico poco attento alle novità.

È il caso delle nuove tecnologie. L'epoca in cui viviamo si sta facendo notare per la forza e per la rapidità con cui si diffondono le conoscenze.

L'informatica come scienza dei linguaggi adatti alle nuove tecnologie si porta dal livello di conoscenza universitaria specialistica per pochi al livello di conoscenza di massa per molti.

Si pone il problema di come insegnarla.

L'elaboratore è una macchina non solo fortemente versatile, ma anche dotata di elevata efficacia visiva e comunicativa perché capace di fornire immagini in movimento e parti scritte.

Sorge l'idea di utilizzare queste capacità di penetrazione dell'elaboratore come sussidio per l'insegnamento.

I due ordini di potenzialità, quello legato all'uso dell'informatica delle nuove leve e quello legato all'uso dell'informatica nella formazione e nell'istruzione di base, stanno però nascendo in Italia in ambienti spesso troppo poco dotati di competenze in campo educativo e didattico con tutti i difetti che prima sono stati enunciati.

Ecco da dove nasce l'attuale pubblicazione.

Essa si pone come contributo al collegamento tra la programmazione didattica in corsi di studio e l'uso del computer.

La nostra intenzione è quella di mostrare, con esempi concreti forniti da esperti in campo educativo e didattico, quali valenze formative siano connesse con la conoscenza e con l'uso di questo nuovo strumento e del suo linguaggio.

Per non dare tuttavia l'impressione che la raccolta di esempi sia asistemica e, preoccupazione ancora più significativa, per consentire a tutti i docenti (quelli alfabetizzati all'informatica, quelli che alfabetizzati non lo sono ancora ma lo saranno presto e quelli che non lo saranno mai) di inquadrare i vari possibili impieghi dell'informatica e degli elaboratori in didattica, si ritiene opportuno fornire i criteri che hanno guidato l'equipe di Risorse nel suo lavoro di studio e raccolta dell'esistente, di confronto con le necessità concrete della scuola di base in Italia.

Dal punto di vista pedagogico/didattico devono essere tenuti nel giusto rilievo sia lo studio che la possibilità di utilizzo delle nuove tecnologie in relazione soprattutto a nuove valenze formative che esse promuovono.

Non possono poi essere tenute in minor conto le implicazioni educative/istruttive connesse con la conoscenza e l'uso dei linguaggi artificiali.

Così pure bisogna aver chiaro, nell'esprimere l'esigenza generale che si introduca l'informatica nell'insegnamento, che esistono tre intenzionalità possibili ben distinte:

l'una tendente a promuovere delle competenze specifiche attraverso l'alfabetizzazione informatica e successivi approfondimenti e ampliamenti, l'altra tendente a promuovere l'uso di software confezionato apposta per "assistere" su temi specifici l'alunno in fase di apprendimento (il cosiddetto CAI), l'altra ancora tendente a utilizzare il computer come sussidio progredito nella gestione dell'organizzazione scolastica e delle strategie didattiche.

1.1.1 La padronanza del progresso tecnico nella formazione di base

In tutte le fasi di intensa trasformazione ed innovazione tecnologica, specie quando i ritmi dei processi si fanno convulsi ed i campi di applicazione della innovazione si fanno così estesi da far ricorrere al termine di "rivoluzione tecnologica", al mondo ed agli uomini della cultura spetta il compito di interpretare, di mediare socialmente l'incremento di sapere scientifico e tecnico che si realizza in tali passaggi della storia, di ricostruire un "senso comune" ed una scala di valori che tali processi tendono a disarticolare.

Almeno tale è il compito che si propone alla cultura "democratica", a partire dalla convinzione che il controllo del progresso tecnico rappresenti, in queste più che in altre fasi della storia dell'uomo, il nocciolo della democrazia.

Di fronte alla cosiddetta "rivoluzione microelettronica" il nostro atteggiamento non può essere diverso: le tecniche e gli strumenti della microelettronica hanno infatti potenzialità innovative paragonabili a quelle che ebbe l'invenzione della carta stampata, mentre lo sviluppo della comunicazione umana che esse consentono ha il medesimo contenuto di progresso che è implicito nel controllo di una nuova forma di energia.

Potenzialità di sviluppo e di progresso, incremento della potenza del lavoro umano, non sono però risultati "fisiologici", per se determinati; ma solo l'effetto combinato di progresso tecnico e di controllo sociale adeguato ad esso; dunque, innanzitutto, effetti dell'allargamento delle basi sociali del sapere scientifico e tecnico da un lato, e della ricostruzione paziente e laboriosa di un senso comune adeguato alle nuove tecnologie dall'altro.

Obiettivi culturali; perciò; o meglio di politica culturale: non fu dissimile l'intenzione dei "philosophes" nel secolo dei lumi, nè di diverso significato il loro legarsi ad iniziative editoriali di divulgazione quale fu l'enciclopedia.

Ma se gli obiettivi di una cultura democratica possono essere descritti con le medesime proposizioni generali del passato, notevolmente diverse sono le condizioni di contesto per il perseguimento di tali obiettivi, sia per i caratteri propri di "questa" rivoluzione tecnologica, sia per gli strumenti sociali che si offrono alla politica culturale.

Tra le caratteristiche più citate della "rivoluzione microelettronica" vi sono la grande velocità di innovazione, e la grande pervasività. I suoi prodotti si moltiplicano e si diversificano rapidamente, rendendo difficile tenere il passo, anche solo in termini di conoscenza generale, dei risultati e delle applicazioni, ma anche si diffonde in tutti i campi, dalla produzione materiale alla vita quotidiana.

Quest'ultimo aspetto è per molti versi ancora più clamoroso del primo: gli "oggetti" della rivoluzione microelettronica e, sia pure con minore immediatezza, le tecniche di utilizzo degli stessi, tendono a pervadere profondamente molti aspetti della vita quotidiana offrendosi al consumo come prodotti di massa.

Le difficoltà di interpretazione culturale sono perciò notevoli: le potenzialità innovative disaggregano modelli consolidati di produzione, di consumo, di costume, contemporaneamente e con equivalente capacità di penetrazione in tutti i settori.

Queste difficoltà scontano anche retaggi della cultura tradizionale, come la separazione tra sapere scientifico e cultura umanistica, specialmente dove essa è più radicata, come nel modello culturale dominante del nostro paese; ma c'è di più.

Le cosiddette nuove tecnologie sono infatti in grado di incorporare su un supporto materiale di notevole potenza e flessibilità, e con bassi costi di produzione, una parte almeno delle prestazioni e dei caratteri che siamo soliti attribuire tradizionalmente al lavoro intellettuale.

Vi è dunque, in parallelo ai fenomeni descritti, una innovazione profonda delle condizioni e degli strumenti della produzione intellettuale, cui per altro spetterebbe innanzi tutto il compito di interpretare e “leggere” la “rivoluzione tecnologica”.

Non è certo caratteristica esclusiva di questa tecnologia il proporsi come modello di “ragione”; per tutte le tecnologie si è sempre realizzata la fusione tra modello culturale e “macchinismo”, tra “razionalità” e tecnica; tuttavia in questo caso la fusione con il supporto materiale si offre a tale livello di completezza che questa tecnologia si propone come “tecnica della ragione”.

Sono diverse dal passato, dunque, sia la scala dei compiti che si propongono alla cultura ed alla sua organizzazione, sia la qualità “filosofica ed epistemologica” della interpretazione culturale di questa rivoluzione tecnologica.

Da ogni versante si affronti la questione, sia a partire dalle trasformazioni nella produzione materiale, sia dalla innovazione trasferita nella vita quotidiana, la scuola ed il sistema formativo sono profondamente coinvolti nelle loro stesse finalità essenziali: quanto a dire la definizione di contenuti e profili professionali, che deve misurarsi con una imponente e radicale riclassificazione del lavoro, e la produzione del sapere e della cultura su basi sociali progressivamente allargate.

Del resto, nonostante le “dimenticanze” e le indifferenze politiche, le delusioni o le incertezze seguite agli anni dello sviluppo e della iniziativa entusiasta, proprio la “crisi” dell'innovazione tecnologica ripropone la formazione, l'istruzione, dunque la scuola ed il sistema formativo di massa, come l'asse moderno della riproduzione del sapere e della cultura e l'ammortizzatore sociale dell'innovazione.

Senza una iniziativa di formazione di massa la “rivoluzione tecnologica” non sviluppa pienamente le sue ricchezze, ma rifluisce creando nuovi linguaggi separati, nuove caste, nuovi riti e sacralità: le sue potenzialità liberatrici si trasformano in poteri più oppressivi e lontani, in aristocrazie più esclusive, in disuguaglianze più radicali.

L'impatto delle nuove tecnologie con l'istruzione e la formazione è un fatto che certo non richiede dimostrazioni: non fosse altro perché queste ultime costituiscono un grande mercato in grado di alimentare con continuità la domanda di hardware e di software.

Il problema, in gran parte inesplorato in Italia, è il modo di organizzare questo impatto, di qualificare la domanda, di dare dignità di “consumatore collettivo” al mondo della scuola e della formazione, di utilizzare la sua stessa valenza economica come leva per indirizzare e governare l'offerta.

I richiami ai compiti di una cultura “democratica” non potrebbero essere completi se non indicassimo anche la necessità di un esercizio collettivo di organizzazione e di produzione di cultura, dentro ed attraverso l'organizzazione del sistema formativo di massa.

Il semplice processo di diffusione tecnologica può anche essere affidato al mercato, ma in tal modo cambia segno e valore, e soprattutto risultati, rispetto agli obiettivi indicati.

1.1.2. Dai “cento fiori” a un possibile piano di politica scolastica: tre principi di fondo

Negli ultimi due-tre anni la pressione del mercato dell'informatica sulla scuola si è progressivamente accentuata; parallelamente gruppi di insegnanti più sensibili hanno effettuato singolarmente o collettivamente tentativi di sperimentazione innanzitutto di auto-alfabetizzazione per iniziative spontanee e diffuse.

Sul complesso degli insegnanti e degli utenti della scuola tutto ciò ha avuto il carattere di un vero e proprio “bombardamento” di opinione, amplificato e rinforzato dalla sensibilità al tema da parte dei mass-media.

Si è trattato e si tratta di un processo di grande interesse e di grandi potenzialità, nel quale sono scese in campo fantasia, creatività, energie autonome del mondo della scuola (accanto ad interessi commerciali) misurandosi con tentativi di varia ispirazione e finalità, senza che queste energie trovassero coordinamento in un quadro di sufficiente omogeneità di criteri e di obiettivi.

Come in tutti i processi “a cento fiori” viene il momento in cui queste stesse energie suscitate richiedono, per non estinguersi, un intervento programmatore, di direzione e di orientamento.

Un “piano nazionale” per l'informatica e la microelettronica nella formazione per tutti i livelli di scolarità, in analogia a quanto realizzato in Gran Bretagna o in Francia, è tutt'ora un problema irrisolto nel nostro Paese: il progetto presentato recentemente, più che discutibile per molti aspetti, è limitato al biennio della secondaria superiore. Occorrono invece, data la maturità dei tempi, indicazioni chiare di merito e di contenuti: è un problema di politica culturale e scolastica che esula da queste note. Se ne facciamo riferimento è perché siamo costretti a postularlo come sfondo alle nostre argomentazioni ed al tentativo di enucleare criteri, principi e contenuti per la diffusione del computer e dell'informatica nella scuola.

Per intanto vorremmo proporre alcuni principi essenziali, misurati nella stessa fase di esperienza “a cento fiori”, che ci sembrano coerentemente congiungere il richiamo alle nuove tecnologie con l'esigenza di inserire la sperimentazione di queste in un disegno organizzato di politica culturale.

Per ciascuno di essi cercheremo di esplicitare conseguenze e implicazioni, almeno le essenziali, di politica scolastica, con l'avvertenza che il riferimento costante sarà alla formazione di base; per i livelli di scolarità più vicini al mercato del lavoro e con preoccupazioni di professionalizzazione, i termini del problema sono infatti notevolmente diversi, anche se non indipendenti.

Il primo enunciato sembra avere la forza dell'ovvietà: occorre proporre e considerare il computer come uno “strumento di lavoro”, un mezzo di produzione con caratteristiche proprie e potenzialità, che vanno esplorate a fondo, per produrre didattica.

Le implicazioni sono un po' meno ovvie: sottolineare il carattere di strumento di produzione significa porre contemporaneamente i problemi di professionalità degli operatori della formazione, e dei modelli di organizzazione del loro lavoro.

Professionalità degli insegnanti rispetto al computer come strumento di produzione a loro affidato, e non solo come “sussidio” o, al peggio, come sostituto di competenze, significa padronanza individuale e collettiva non solo dell'uso dello strumento, ma delle potenzialità formative delle tecnologie di utilizzo.

Si tratta di uno spettro ampio di competenze, che vanno dalla produzione sia individuale che collettiva del software didattico e di gestione della didattica, alla scelta critica ed intelligente del software prodotto altrove. Certo è che non ci si può accontentare di una prospettiva nella quale gli insegnanti siano utilizzatori passivi di una macchina per una didattica determinata altrove.

Può sembrare troppo ambizioso ed irrealistico proporsi la diffusione di massa delle conoscenze di tecniche di programmazione e di almeno un paio di linguaggi, come tratti specifici di professionalità per tutti gli insegnanti; d'altro canto per formare un buon programmatore occorre circa un anno di formazione, molto di più per formare un buon insegnante. Dunque contrariamente a quanto può apparire è più facile dare buone competenze di programmazione ad un "tecnico" della didattica, che non trasformare un tecnico della programmazione in un professionista della didattica.

È una verità dimostrata anche dallo scarso valore didattico che ha (per ora) il software in circolazione.

Fino a che le competenze di "softwaristi" e di "didattici" non si ricongiungono (ed è un processo che va organizzato ed è parte consistente delle ragioni del richiamo ad un piano nazionale) l'insegnante che si limiti ad essere un "diskettaro" non solo interpreta un ruolo professionale mortificato ed inaccettabile, ma non sviluppa e promuove nessuna crescita di questa tecnologia, e contribuisce a limitarne le potenzialità formative ed a esaltare le sole convenienze del mercato (o meglio, alcune convenienze rispetto ad altre).

Padronanza collettiva di questi "strumenti di produzione" significa infine affrontare il tema ostico della organizzazione del lavoro formativo. È chiaro infatti che tempi, spazi, cadenze del lavoro scolastico tradizionale non possono rimanere inalterati con l'introduzione di tecnologie: si pensi che gli involucri spazio-temporali si rivelano inadeguati a contenere un lavoro arricchito di strumenti tradizionali come il libro, quando questo sia utilizzato in modo esteso, e non nella eccezione limitata del libro di testo.

Poiché il rapporto tecnologia-organizzazione è tutt'altro che univocamente e oggettivamente determinato, ma dà spazio alla decisione ed alla scelta "politica", si tratta di affidare quest'ultima alla competenza collettiva dei "produttori" di didattica e non alla presunta oggettività del "macchinismo", e tanto meno alla autodifesa corporativa.

Il secondo assunto, già citato, può essere così espresso: è necessario che la diffusione di massa degli strumenti e delle tecniche dell'informatica discenda da un progetto finalizzato che determini priorità e coerenze.

Gli indirizzi essenziali dovrebbero essere:

1. Lo sviluppo della ricerca: è singolare che la ricerca applicativa dell'informatica sia sviluppata ed in progresso in tutti i più svariati campi, e rimanga invece defilata e subordinata proprio per le applicazioni formative.

Il termine ricerca è però qui utilizzato in senso molto lato, comprendendo sia quella indirizzata alle intelligenze artificiali ed ai sistemi esperti, sia la sperimentazione nella formazione di metodologie avanzate (quindi con applicazioni limitate e "pilota"), sia la ricerca sull'hardware e le sue caratteristiche ottimali per un impiego scolastico, sia quella sui linguaggi.

Tutto ciò ha già i suoi momenti organizzati ed i suoi presidi nell'università e nei centri di ricerca come il CNR; manca invece di un coordinamento complessivo e, soprattutto, di un campo generalizzato di applicazione, che verifichi indirizzi e finalità e retroagisca determinando convenienze e stimoli.

2. Un piano di alfabetizzazione: continua ad essere un obiettivo prioritario nonostante le esperienze spontanee degli ultimi anni.

Potremmo sinteticamente esprimerlo con l'obiettivo di allargare a tutti gli insegnanti la conoscenza operativa di almeno due linguaggi di programmazione, come tratto specifico di professionalità.

3. La determinazione di quote progressivamente più ampie di sperimentazione in classe. L'adeguamento culturale dei docenti alle "nuove tecnologie" ha senso solo se è in parallelo ed al contempo lo strumento per una vasta sperimentazione "sul campo" (e quindi "in classe", nel fare scuola quotidiano, trasversale alle singole discipline).

D'altro canto l'obiettivo di elaborare una cultura non subalterna della "rivoluzione

tecnologica" non può che essere "sperimentale" e coinvolge la formazione culturale di base, non solo progetti "pilota".

4. L'organizzazione stabile e permanente della socializzazione delle esperienze. In questo campo ciò deve significare la creazione di una "banca del software didattico" nella quale non ci si limiti a conservare e memorizzare: occorre invece un sistema organizzato di *raccolta, validazione, riproduzione standard, circolazione del software*, commissionato e prodotto e sperimentato dagli insegnanti stessi.

Gli indirizzi sinteticamente richiamati per un piano nazionale (o anche solo per muoversi in modo organizzato all'interno di tumultuosi processi spontanei, nei quali è difficile rintracciare criteri unificanti) hanno conseguenze di grande portata nella scelta dell'hardware e dei linguaggi.

Si tratta infatti di diffondere in grande copia computer in tutte le scuole e di perseguire criteri di sufficiente omogeneità per rendere al meglio comunicabili e scambiabili le esperienze.

L'hardware innanzitutto: l'impostazione che stiamo tentando di esplicitare vorrebbe essere largamente indipendente ed immune dall'empirismo della "macchina"; ma è un fatto che pensando alle dimensioni del mercato scolastico, nonché al privilegio accordato alla formazione di base, nonché alla sottolineatura dei valori formativi e cognitivi rispetto ad approcci strumentali, si identifichino anche alcune caratteristiche delle "macchine" sulle quali, anche solo in termini di ragione pratica, è necessario puntare l'attenzione.

Si tratta di compiere un grande sforzo economico, che non può che essere guidato in tal caso dall'esigenza di investimenti a basso costo unitario e molto diffusi; si tratta di rispondere ad una grande pervasività dell'informatica nella vita quotidiana con l'uso formativo di oggetti che entrano nel consumo di massa; si tratta di suscitare competenze "produttive" combattendo tendenze subalterne al consumo; dunque macchine "piccole", di basso costo unitario, ma in grande numero nelle classi; la potenza o la raffinatezza delle prestazioni devono cedere il passo alla flessibilità ed alla facile programmabilità.

Per quanto riguarda i linguaggi, si è sviluppato un interessante dibattito tra i sostenitori della migliore valenza formativa di questo o quel linguaggio: a tutt'oggi pare consolidata la scelta del LOGO come linguaggio adatto alla alfabetizzazione informatica della scuola dell'infanzia, mentre i costrutti logici più ricchi del PASCAL vengono contrapposti alla larghissima diffusione e semplicità del BASIC; non è facile dirimere una questione che non può essere solo teorica, ma che ha grandi aspetti "pratici" di organizzazione della politica scolastica e culturale.

È solo per motivi di "ragion pratica" infatti che ci sentiamo di indicare il BASIC come linguaggio di una generale alfabetizzazione da realizzare nella scuola dell'obbligo: non solo è il linguaggio implementato nella stragrande maggioranza delle macchine che hanno le caratteristiche ricordate, ma la gran parte delle energie disponibili per un "piano nazionale" di grande portata (leggi formatori) è in realtà "BASIC-orientata".

Certo la posizione più coerente con le premesse ed i principi che stiamo sviluppando sarebbe quella di invertire i termini della questione: dalla determinazione degli standard necessari sia dell'hardware che dei linguaggi, farne discendere un impegno "produttivo" ad hoc per la formazione. È quanto è stato fatto per esempio in Francia con la mobilitazione finalizzata della produzione di hardware nazionale e la creazione di un linguaggio specifico (che non a caso ha caratteristiche che potremmo porre a metà tra il Basic ed il Pascal).

Ma il tempo non trascorre mai invano, e le scelte più valide ieri, ripercorse oggi, rischiano solo di tardare la realizzazione di progetti che sono invece urgenti.

L'insieme degli obiettivi di un progetto finalizzato potrebbe essere dunque così sintetizzato: che tutti gli insegnanti abbiano una base conoscitiva di BASIC e LOGO tale da

poter utilizzare sperimentalmente il grande numero di piccoli elaboratori diffusi nelle varie classi grazie ad una politica di investimenti a basso costo unitario. A monte di questa sperimentazione diffusa, una ricerca coordinata che si alimenti dei risultati di quella e a sua volta ne verifichi, modifichi, raffini gli indirizzi e gli strumenti. A valle un sistema organizzato di validazione, standardizzazione, circolazione del software prodotto “sul campo”.

Non ci paiono esistere molte alternative al rischio di degenerare questa fase ricca di spontaneità e di creatività in una babele delle esperienze e relativa mortificazione delle potenzialità.

Il terzo ed ultimo “enunciato” si riferisce ad un principio essenziale che deve guidare l’insieme della sperimentazione e della diffusione delle nuove tecnologie nella didattica: la definizione delle strategie didattiche costituisce la competenza propria, individuale e collettiva degli insegnanti, essendo l’asse (da rinforzare e sviluppare) della professionalità.

Nessuna tecnologia può essere sostitutiva di questa competenza (al contrario di altre, come per esempio la comunicazione); ogni tecnologia deve invece dare strumenti a questa competenza, esaltandola, potenziandola e liberandola anche dai molti vincoli, oggettivi e non, che la ripetitività di talune operazioni o il burocratismo, o i difetti di comunicazione, impongono.

In altre parole il campo di attività costituito dalla determinazione degli obiettivi culturali e formativi, dall’adattamento di questi alle condizioni concrete dei singoli discendenti, dalla determinazione delle cadenze, dei compiti, degli obiettivi intermedi, degli standard, è campo “proprio” della professionalità del singolo e del collettivo senza possibilità di sostituzione.

La fase di lavoro scolastico che siamo soliti chiamare “programmazione didattica” non può essere incorporata dal computer e sottratta ai docenti: in questa, più che mai, il computer può essere strumento capace di migliorare la produttività e l’efficacia del lavoro, ma non può, nè deve, implementarne la direzione e le finalità.

Si tratta di una affermazione di un principio che ha congiuntamente valore di salvaguardia e di promozione progressiva di professionalità. Tende cioè a rispondere positivamente alla duplice reazione che le nuove tecnologie suscitano al loro ingresso nel mondo della scuola: da un lato la ripulsione passiva e spaventata, che proviene da una tradizione di lavoro destrumentato e povero di tecnologia, dall’altro l’adesione entusiasta che pensa di affidare all’informatica la soluzione (finalmente) dei tanti e troppi problemi irrisolti, e della sensazione diffusa di inadeguatezza ed impotenza di fronte ad essi.

Tanto il timor panico che l’affidamento mitico sono atteggiamenti di subalternità, altrettanto pericolosi nel promuovere la sacralità degli strumenti e delle tecniche, piuttosto che il loro uso controllato e consapevole.

Crediamo che il riferimento a questi principi di fondo renda più agevole lo sviluppo delle considerazioni che seguiranno sull’uso del computer nella scuola, e più chiaro il senso di alcune scelte che indicheremo.

1.1.3 Le utilizzazioni possibili del computer nella scuola: alfabetizzazione informatica e informatizzazione della didattica

Per comodità ed ordine di esposizione possiamo distinguere quattro campi di utilizzazione (e di utilità) del computer nella scuola: la gestione dell’azienda scuola”, la didattica, la gestione della didattica e la gestione territoriale del sistema formativo.

Il primo aspetto, pur importante, tocca marginalmente l’argomento di queste note ed è anche meno ricco di aspetti problematici.

Da questo punto di vista l'informatizzazione della gestione degli istituti scolastici non presenta "qualità" diverse dalle esperienze ormai consolidate della informatizzazione della gestione aziendale. Anche se la scuola è una "azienda" imperfetta e "sui generis", poiché un bilancio scolastico è diverso da un bilancio aziendale almeno quanto un archivio degli iscritti è diverso da un archivio clienti, i riferimenti e le esperienze sono già molto sviluppati.

Sarebbe invece interessante analizzare i problemi di raccordo tra questo livello di informatizzazione e quello del Ministero della Pubblica Istruzione per la gestione complessiva del sistema scolastico, e l'architettura complessiva che viene ad assumere il sistema in base alle scelte fatte al centro ed alla tradizionale spontaneità lasciata alla periferia.

Anche l'ultimo campo di utilizzazione indicato (la gestione territoriale del sistema formativo) non presenta particolari scogli concettuali.

Si tratta di costruire un vero e proprio sistema informativo delle esperienze e delle occasioni formative sul territorio, il supporto tecnico alla ricostruzione di un modello domanda-offerta di formazione. Quindi di un sistema di archiviazione, conservazione ed erogazione di tutte le informazioni che riguardano le esperienze didattiche, le ricerche, i bacini di utenza, le occasioni formative ecc., che attengono ad un certo territorio.

Si può e si deve discutere e decidere circa le competenze, gli ambienti territoriali, le responsabilità di gestione, l'architettura, di questo sistema; tuttavia la sua creazione corrisponde all'obiettivo di dotare di uno strumento tecnico potente e flessibile una intenzione ormai da tutti predicata: l'integrazione del sistema formativo allargato, che coordini e programmi l'insieme dell'offerta formativa all'insegna della integrazione e del pieno utilizzo delle risorse, pubbliche e non.

La didattica e la sua gestione rappresentano, tra le applicazioni citate, quelle più ricche di problematica e più necessitanti di chiarezze concettuali, ma anche più pertinenti all'argomento di queste note.

Chiediamoci, per impostare il ragionamento, quali siano le prestazioni utilizzabili "didatticamente" di un elaboratore.

In grande sintesi il computer è uno strumento per:

- a) eseguire calcoli;
- b) memorizzare dati e informazioni;
- c) comunicare;
- d) utilizzare le caratteristiche precedenti nella risoluzione di problemi di qualunque grado di complessità, purché abbiano almeno una soluzione.

La somma di queste caratteristiche ne fanno uno strumento potente e flessibile, "trasversale", di formazione pluridisciplinare.

Dunque si tratta di uno strumento didattico non esclusivo di una o qualche disciplina; piuttosto legato ad una didattica del "problem solving", quindi alla sua ricchezza, ma anche ai contorni epistemologici che essa ha, per quanto ampi essi siano, ed alla didattica del "pensiero formale", certo decisiva, ma non esclusiva.

Per quanto schematica, la stessa determinazione dei contorni del campo applicativo nella didattica, e sono amplissimi come si può intendere, mette in luce quali siano per noi gli aspetti formativi che giustificano, nel profondo, l'approccio positivo ed interessato alla introduzione del computer e dell'informatica nella formazione di base.

Dal punto di vista strettamente formativo ci interessano i valori cognitivi connessi con la ricerca, l'invenzione, la determinazione dell'algoritmo risolutivo; e ancora quelli connessi alla traduzione di quest'ultimo in "programma".

La programmazione, e cioè la predeterminazione, la prefigurazione, l'evocazione e quindi il controllo della esecuzione della macchina è l'aspetto "antropomorfo" di questa tecnologia.

La sacralità ed il feticismo legati agli “oggetti” della microelettronica (e quindi la subalternità ad essi) si superano se il primato formativo è assegnato proprio a questi aspetti decisamente ed esclusivamente cognitivi.

D'altro canto la programmazione corrisponde anche ad una tappa decisiva della comunicazione umana: il modo di risolvere e di comunicare la risoluzione di problemi complessi.

Sottolineare e privilegiare questi aspetti fino a proporsi di farli diventare parte integrante della formazione di base, e quindi di massa, significa, d'altro canto, muoversi nella direzione di superare il carattere “specialistico” e cominciare ad intendere l'informatica come una disciplina “umanistica”. Ciò ha per la nostra cultura un forte valore di riunificazione e ricomposizione dei saperi.

A partire da questa impostazione, di privilegio assoluto degli aspetti cognitivi, se ne potrebbe a prima vista concludere che l'uso della “macchina” è accidentale, o addirittura inutile.

Ciò che conta dal punto di vista formativo è la “metodologia”, la ricerca dell'algoritmo, la sequenzializzazione a passi discreti del ragionamento; ma non si devono trascurare gli aspetti più propriamente didattici connessi con l'uso di uno strumento e con il passaggio immediato dalla teoria alla realizzazione pratica, che solo il computer garantisce.

Le teorizzazioni dell'informatica “povera e cognitiva” se ci trovano assolutamente consenzienti circa le motivazioni “formative”, ci paiono perciò riduttive sotto il profilo didattico, per molte ragioni.

In primo luogo l'esaltazione dei valori cognitivi, privata del trasferimento nella prassi, della traduzione concreta del funzionamento di una macchina perde gran parte dell'efficacia didattica, per residuare una pura “ginnastica mentale”, che rischia di trovare nei discenti scarsa motivazione.

In secondo luogo si rischia di perdere così il valore didattico essenziale del diretto legame teoria-prassi che il calcolatore consente: l'immediatezza del risultato. Un esercizio di problem-solving che consente una immediata verifica delle soluzioni, consente di concentrare le energie, che spesso sono disperse nelle fasi operative lunghe e ripetitive, nella fase creativa della individuazione delle soluzioni.

In terzo luogo, sganciata dall'uso concreto di una macchina, la metodologia di cui si sottolineano giustamente i valori cognitivi, rischia di provocare una assolutizzazione del valore del pensiero formale e scapito di altri approcci.

Non è solo questione di “misura e di equilibrio”: senza la macchina “che ne rappresenta la finalizzazione empirica (e quindi i contorni di campo di validità) la “ginnastica mentale” dell'algoritmizzazione rischia di capovolgersi in rappresentazione dei processi mentali, e paradossalmente assumere il ruolo di “latino moderno” (rispetto a quello tradizionale, che parimenti insegna a “ragionare”). Il che rappresenta una singolare mistificazione per chi abbia decisamente sposato il punto di vista “cognitivo”.

Siamo per una informatica povera e cognitiva nella formazione di base, ma proprio per questo pensiamo sia necessario l'uso concreto di una macchina “povera”, che costringa ad esercitare il massimo di competenze e di “astuzie” nelle tecnologie di utilizzo, e mantenga contemporaneamente intatta non solo l'efficacia didattica, ma la finalizzazione empirica di una metodologia di pensiero, impedendo che essa si proponga come esclusiva.

La chiarezza nei rapporti tra valori didattici-cognitivi-formativi consente di porre nettamente una distinzione fondamentale tra “didattica con il computer” e “didattica computerizzata”.

Al di là dell'efficacia della distinzione terminologica, con la prima si identifica l'uso del computer in una didattica della risoluzione dei problemi, tesa, per tutte le discipline, ad

utilizzare lo strumento con un continuo esercizio di competenze nella conoscenza di linguaggi e nella programmazione, alla ricerca ed invenzione dell'algoritmo e della corretta esecuzione della macchina.

Tramite questa didattica si procede alla "produzione" di software proprio (da parte degli insegnanti così come dei discendenti); la raffinatezza formale dei programmi cede il passo al valore della "produzione propria" ed al significato di sperimentazione sul campo; la strategia didattica è saldamente controllata "nel merito e nel metodo" dai protagonisti del lavoro formativo.

Con i termini "didattica computerizzata" vogliamo sottolineare i più diversi utilizzi del computer, nei quali il protagonismo didattico diretto è trasferito alla macchina stessa.

Il software è somministrato e "consumato", non prodotto direttamente, nel processo di apprendimento.

Occorre ricordare che in generale gli usi più diffusi del computer nella didattica appartengono a questa categoria, ma sono anche quelli che suscitano maggiori delusioni e riserve.

La didattica "computerizzata" si sviluppa sostanzialmente in quattro direzioni, che riassumono efficacemente tutte le varietà d'uso del computer nella formazione:

- a) Come insegnante; il computer somministra direttamente allo studente unità didattiche, intervallandole a test diagnostici, in base ai quali vengono effettuati rimandi di recupero e graduata la sequenza delle unità e la velocità del procedimento.
- b) Come esercitatore; il computer somministra esercizi a partire da un livello di complessità predefinito, tramite un test di ingresso, e via via crescente in base alla misura delle prestazioni del discente.
- c) Come simulazione; l'efficacia della rappresentazione video viene utilizzata per presentare graficamente e con animazione i più diversi fenomeni fisico-chimici. Il discente può interferire modificando i parametri dei modelli proposti e misurandone le conseguenze e gli sviluppi.
- d) Come gioco; non si tratta dei "videogame" ma della ricostruzione al computer di situazioni e "storie" senza legami con la realtà, ma che richiedono all'utente una serie di decisioni la cui meccanica e logica ha attinenza con possibili decisioni reali.

Questi modelli di didattica computerizzata, che nell'uso corrente spesso combinano le loro modalità, hanno in comune il fatto che il software da cui prendono origine è generalmente esterno al processo di apprendimento, prodotto altrove e somministrato.

Il discente è subordinato alla macchina e può al massimo regolarne le cadenze (o spegnere il tutto naturalmente), ma è sottoposto ad una strategia predeterminata e con un bassissimo (o nullo) grado di adattabilità.

Poiché il grado di efficacia didattica è in questo caso in gran parte affidato alla qualità formale ed alla "raffinatezza" del programma, nella maggioranza dei casi si tratta di software prodotto da case specializzate.

Ciò significa che anche il docente si trova spesso ad essere "consumatore", che può scegliere ma nell'ambito di un repertorio prefissato, e sul quale non può intervenire in termini di adattamento.

Ci muoviamo quindi, nel caso di tali utilizzazioni, ai confini, se non all'esterno, dei "principi" precedentemente indicati come "guida" all'uso del computer nella didattica. Tuttavia non vorremmo che l'aver abbinato la distinzione tra didattica con il computer e didattica computerizzata, alla distinzione tra "produzione" e "consumo", facesse trarre conclusioni negative pregiudizialmente circa la legittimità e la funzione che può avere l'uso dell'elaboratore anche secondo queste ultime modalità.

Si tratta di conoscere esattamente le caratteristiche dei mezzi e delle tecniche a disposizione e di utilizzarle coerentemente ai propri scopi, senza confondere gli uni con gli altri,

ma anche senza assolutismi o pregiudiziali: il lavoro didattico richiede sempre un insieme di interventi nel quale c'è posto per molti strumenti e per tecniche diversificate.

Vorremmo solo aggiungere qualche considerazione circa la "simulazione", che sembra essere una modalità di uso didattico del computer in grande sviluppo. Si tratta di una tecnica didattica di grande efficacia: alcuni fenomeni sono solamente "simulabili" e non ricostruibili in laboratorio; dunque ben venga uno strumento così potente di simulazione che tra l'altro consente di assegnare ai parametri con cui si lavora valori reali, ciò che quasi mai accade nelle simulazioni di laboratorio più tradizionali.

La diffusione della simulazione è però anche legata al fatto, molto "empirico", che ormai un computer costa meno di qualsiasi laboratorio di fisica e di chimica, e che quindi risulta agevole sostituire integralmente questo con quello.

Va ricordato, allora, che dal punto di vista formativo nulla sostituisce l'interazione diretta "materiale" con la realtà, e che tutti noi, ma in particolare i giovani, viviamo immersi in un mondo di informazione "mediata", che rischia quindi di scambiare in continuazione la realtà con le sue rappresentazioni "simulate".

A costo di fare esperimenti meno brillanti, è meglio "farli" che "simularli". È ovviamente anche in questo caso "questione di misura".

Le modalità di didattica "computerizzata" citate rappresentano, con la ricchezza che lo stesso progresso tecnico comporta, varietà interne alla filosofia più tradizionale e consolidata del CAI.

Quest'ultima si fonda sulla somministrazione di una strategia didattica, scomposta in obiettivi e contenuti elementari, attraverso un percorso a "tappe" nel quale è inserito il discente, in cui cadenze e velocità di scorrimento sono legate a "prestazioni" misurate ad intervalli attraverso test ed esercizi.

L'impostazione è nettamente "skinneriana" e il computer dà ad essa la potenza di un mezzo tecnico particolarmente adeguato ad interpretarla.

La raffinatezza tecnica ha permesso di presentare percorsi formativi più articolati e flessibili, ramificati e in parte "plasmabili"; ma non è la tecnica che qui ci interessa per un approccio critico, bensì la filosofia ispiratrice.

Proprio perché la tecnica non è "neutra" ma strettamente legata al modello di direzione ed al modello culturale nel quale si inserisce; è da qui che occorre partire.

La scomposizione dei contenuti dell'apprendimento in obiettivi elementari, il loro realizzarsi in prestazioni misurabili, la somministrazione in sequenze intervallate da meccanismi ricompensa e di rinforzo (tenuto conto delle caratteristiche "autovalorizzanti" dell'animale umano) rappresentano solo espressioni tecniche coerenti ad una filosofia fondata sulla predeterminazione della strategia formativa all'esterno, e in certo modo indipendentemente, dal contesto dell'apprendimento.

Il calcolatore fornisce a questa predeterminazione "esterna" il supporto tangibile di una incorporazione su un substrato materiale capace di cristallizzare e rendere efficientemente operanti decisioni e direzioni lontane ed estranee a tutti i protagonisti del processo.

È evidente qui l'impatto negativo e radicale con quello che abbiamo precedentemente enunciato come "terzo principio": la necessità di far permanere le decisioni e le scelte di strategia di didattica come contenuto essenziale della professionalità del docente.

Ma è un impatto "di principi" che tende ad escludere e a salvaguardare da qualsiasi "esproprio" di professionalità, ed anzi ad arricchirla nella misura in cui si arricchiscono gli strumenti in dotazione. Non è *immediatamente* un impatto tra "tecniche".

Il CAI ha suscitato e suscita drastiche ripulsioni, soprattutto laddove si guardi con grande e fondata sensibilità al peso determinante delle "variabili di contesto" nei processi formativi, al ruolo che vi devono svolgere il consenso e la partecipazione, la decisionalità e la creatività dei protagonisti, nonché all'influenza, per molte ragioni essenziale, degli aspetti relazionali ed effettivi nell'apprendimento.

La questione fa capo ad un dibattito così radicato (e per quanto ci riguardi risolto) nel cuore della scienza pedagogica che non riteniamo utile soffermarci più a lungo.

Tuttavia è innegabile l'efficacia pratica che metodologie che si rifanno più o meno esplicitamente al CAI hanno, quando siano applicate a certe fasi della formazione, scolastica e non (si pensi a quanto spesso ad esse si ricorra nella formazione e nell'addestramento aziendale). La ripulsa di principio si rivela debole nella realtà se la contrapposizione "filosofica" induce a "liquidare", senza approccio critico adeguato, anche la "tecnica".

È sempre possibile recuperare una tecnica, nata in un contesto culturale e "filosofico", entro un contesto diverso, salvandone "utilitaristicamente" l'efficacia pratica, e modificandone criticamente i contorni del campo di applicabilità, ed i significati.

Del resto in questi anni nella scuola italiana una operazione analoga (recupero di una "tecnica" entro una mediazione culturale diversa da quella di origine) è stata fatta per esempio con il Mastery Learning semplicemente spostando il centro di gravità di un modello molto formalizzato di processo di insegnamento e di apprendimento, sulle fasi della programmazione e definizione degli obiettivi didattici, ed impegnandovi programmaticamente la professionalità, le partecipazioni ed il consenso dei protagonisti del processo formativo.

Analogamente si può procedere con la didattica computerizzata, che fa riferimento più o meno diretto al CAI.

Innanzitutto con una rigorosa delimitazione del campo di applicazione: in qualunque processo di insegnamento-apprendimento vi sono fasi ripetitive, esercitative, routinarie, o di semplice addestramento.

Se è saldo e chiaro l'assunto di quello che abbiamo indicato come "terzo principio", se quindi è piena la padronanza della strategia didattica che si vuol perseguire, non si vede perché rinunciare all'apporto di maggiore efficacia del lavoro che tale metodologia dà se applicata a queste fasi.

In secondo luogo, la didattica computerizzata può consentire (alle medesime condizioni di controllo del processo) una diversificazione del linguaggio e degli strumenti didattici, che occorre spesso utilizzare gli uni accanto ed insieme agli altri, per rispondere con un insieme di interventi ad una domanda formativa spesso molto diversificata, addirittura da individuo ad individuo.

In questi casi, come è ovvio, la prima misura di un efficace controllo della strategia didattica da parte delle competenze professionali dei docenti sta proprio nella scelta del software, tra una offerta molto varia e che richiede competenze non solo didattiche per l'orientamento. Sia detto di passaggio, è questo un motivo in più che ci convince della correttezza dell'obiettivo di costruire competenze "produttive", che sono contemporaneamente la massima garanzia di un "consumo" intelligente e consapevole.

In terzo luogo non è da escludere una coniugazione più stretta tra il nostro "terzo principio" e la didattica computerizzata, nella ipotesi di una costruzione "in proprio" del CAI da parte di ciascun docente.

In sostanza non è irrealistico pensare di superare la "rigidità" inaccettabile di un rapporto una volta realizzato, rendendone flessibile al massimo la fase di realizzazione, affidandola alle capacità "produttive" di ciascun docente.

In termini di principio sarebbe in tal modo salva la "adattabilità" del prodotto alle numerose variabili in gioco nell'insegnamento-apprendimento.

Non sembri ozioso delineare anche questa prospettiva: i primi tentativi di produrre didattica computerizzata da parte dei docenti, che in questi anni hanno più attivamente partecipato alla fase spontanea di sensibilizzazione ed alfabetizzazione, sono nonostante tutti, riconducibili a metodologie CAI.

Si è già detto che a differenza che nella opzione "didattica con il computer", che

privilegia gli aspetti cognitivi nell'uso dell'elaboratore in classe, la "didattica computerizzata" fonda gran parte della sua efficacia sulla qualità formale del prodotto, sulla sua "appetibilità" è quindi sulla raffinatezza dei programmi.

Su questo piano non c'è concorrenza possibile del "fai da te" con il software specialistico prodotto "altrove", a meno che non si ipotizzi la creazione di un "metalinguaggio" che consenta al docente, anche con piccole conoscenze di programmazione e di linguaggi appropriati, di produrre le proprie "pagine didattiche" al calcolatore sfruttando capacità grafiche, di animazione, di simulazione di macchine (a questo punto) più complesse di quelle ipotizzate per una diffusione di massa.

Linguaggi, o meglio metalinguaggi, di questo tipo sono già offerti sul mercato da case produttrici di hardware (qui sarebbe forse più corretto parlare di programmi di utilità indirizzati alla didattica): con le premesse di principio da cui ci si è mossi in queste note è evidente che consideriamo anche questa prospettiva carica di tali significati di politica scolastica, che la creazione di un vero e proprio metalinguaggio didattico non può essere il prodotto che di interessi generali e di una ricerca didattica appropriata; comunque non il risultato dell'esplorazione delle potenzialità di una macchina pensata e prodotta soprattutto per altri scopi.

Per concludere: il privilegio assegnato alla "didattica con il computer" ed agli aspetti cognitivi che vi sono sottesi non può significare il rifiuto pregiudiziale della "didattica computerizzata". Una volta consolidati i principi ispiratori di una scelta, la diversificazione delle metodologie e degli strumenti deve ubbidire a criteri di critica e razionale individuazione delle fasi di utilizzazione e delle modalità di applicazione, senza timori infondati e senza altrettanto pericolosi entusiasmi acritici.

Una volta individuate le diverse modalità d'uso del computer nella didattica ci si può interrogare sulle fasi del lavoro scolastico nelle quali il computer trova la sua collocazione ottimale.

La risposta non è semplice perché caricata di un notevole grado di ipoteticità: fino ad oggi non è maturata esperienza sufficiente a porre in modo pieno e corretto problemi di riorganizzazione del lavoro nella scuola. Le esperienze condotte in questa fase sono state soprattutto fondate sulla disponibilità individuale o di piccoli gruppi, quindi non ancora influenti sulla organizzazione dei processi.

Partiamo quindi da un modello ipotetico di organizzazione del "ciclo produttivo della didattica" per individuarne i "luoghi" di più opportuna inserzione dell'elaboratore; la sua rappresentazione è nello schema a blocchi della fig. 1.1.

Il "ciclo" è sostanzialmente composto di:

- 1) una fase "a monte" di progettazione: individuazione degli obiettivi e delle strategie, determinazione dei contenuti delle unità didattiche;
- 2) una sequenza lineare costituita dalla somministrazione ed elaborazione delle unità didattiche stesse;
- 3) inserzioni trasversali costituite dalla valutazione diagnostica e dalla eventuale somministrazione di unità didattiche di recupero;
- 4) una diffusa attività di testing.

Dal simbolismo dello schema risulta chiaro che, coerentemente a tutta la trattazione precedente, la "computerizzazione" non può riguardare la fase di "progettazione" e di programmazione, almeno nel senso che essa non può essere trasferita al computer; mentre il computer può essere utilizzato come strumento utile per la "gestione" della fase (documentazione, memorizzazione, ecc: vedi oltre).

La sequenza lineare dello svolgimento delle unità didattiche è invece il luogo di più

ampia utilizzazione del computer secondo tutte le modalità esaminate: in particolar modo per la costruzione di programmi nella didattica del “problem solving”, capace di suscitare ed utilizzare direttamente competenze “informatiche” dei docenti e dei discenti.

L’elaboratore può diventare particolarmente prezioso nelle fasi trasversali del recupero, sia per la diversificazione dei linguaggi didattici che consente rispetto alla sequenza principale, sia per la gestione contemporanea di spazi didattici individualizzati (che più spesso è un problema irrisolto, nonostante sia predicato, nella organizzazione didattica “normale”).

Tra le altre sue “qualità” l’elaboratore si presenta come un interlocutore senza grande fantasia, ma testardo e paziente, pronto a ripetere all’infinito, e a correggere continuamente.

Lo snodo tra sequenza principale e fasi trasversali richiede accorgimenti organizzativi, ridistribuzione di spazi e di tempi di lavoro, ma anche tecnologie diversificate nella ricerca della comunicazione didattica ottimale alle caratteristiche individuali per realizzare la promozione più ampia possibile.

Infine il testing, sia per la somministrazione, sia soprattutto per il trattamento delle risposte, trova ovviamente grande ausilio in un supporto tecnico che consente di semplificare e sveltire il trattamento di notevoli quantità di dati.

Sotto il profilo della organizzazione strumentale siamo ancora in una fase assolutamente iniziale: in questi anni le scuole, nelle quali hanno agito gli insegnanti più sensibili e attenti al nuovo, si sono spesso dotate di piccoli laboratori di informatica, utilizzando piccoli elaboratori in numero sufficiente per lavorare “con la classe”; più spesso l’insufficienza di orientamento, o semplicemente l’efficacia della sponsorizzazione di alcune case di hardware, hanno finito con l’introdurre nella scuola un elaboratore, magari prestigioso, ma insufficiente ad un vero e proprio lavoro con gli studenti. Una sorta di “monumento alla modernità possibile”.

È possibile oggi pensare di superare anche la dimensione del “laboratorio” di informatica, che ha interpretato in questa fase il particolare interesse che gli insegnanti di alcune discipline (sostanzialmente educazione tecnica e matematica) hanno sviluppato; l’inserimento “orizzontale” del computer nella didattica quale è rappresentato nel nostro schema fa pensare piuttosto ad un’aula informatizzata, la cui costruzione è resa relativamente semplice dal fatto che anche i piccoli elaboratori “domestici” possono oggi lavorare in rete.

Potrebbe anzi essere proprio questo il passo successivo da compiere, per misurare anche le conseguenze sul piano dell’organizzazione del lavoro di alcune scelte culturali e tecnologiche che si sono affermate nella fase “a cento fiori”.

Nel modello che abbiamo ipotizzato, la didattica si trova al centro di una serie di servizi alla sua gestione, che possono senza dubbio essere informatizzati; anzi, solo l’informatizzazione li rende possibili nella condizione di dispersione territoriale dell’unità scolastica.

Abbiamo rappresentato la situazione nello schema 1.2. In esso sono state dettagliate le attività di gestione della didattica sia a monte che a valle del processo didattico diretto, elencando le “prestazioni” che sarebbero richieste per ciascuna fase, ed indicando con il solito simbolismo i luoghi di inserimento del computer.

Lo schema non necessita di commento ulteriore, salvo l’avvertenza che con i termini “attrezzatura didattica” abbiamo inteso indicare i processi di messa a punto e di assetto della “macchina” didattica che precede il lavoro in classe.

Qualche considerazione meritano invece le fasi di documentazione e memorizzazione del lavoro didattico.

Il livello di comunicazione tra i docenti in merito al comune lavoro sono di norma compromessi da due alternative possibili: la ridondanza del dettaglio interpersonale dei

consigli di classe o degli stessi colloqui informali, utilissima per la reciproca conoscenza, ma non riproducibile e quindi legata alla singola esperienza; oppure la formalizzazione estrema delle relazioni burocraticamente elaborate a scadenze fisse durante l'anno, povere di significato ed evanescenti circa le indicazioni accumulabili.

Le conseguenze di questa situazione sono sostanzialmente due: in primo luogo la difficoltà di accumulare i risultati, il che comporta che ogni "ciclo didattico" ricominci sostanzialmente daccapo, utilizzando scarsamente le esperienze precedenti (fatte salve naturalmente le esperienze individuali; ma non di questo si tratta). In secondo luogo non si danno mai misure del lavoro svolto, e quindi verifiche di "produttività".

Costruire un livello organizzato di "comunicazione" (archiviazione, memorizzazione, integrazione) delle esperienze e dei risultati, che si fondi su un supporto tecnico, capace insieme di standardizzare i messaggi e di mantenere ricchezza di significati, avrebbe il duplice scopo di "accumulare" e quindi rendere "esponenziale" lo sviluppo delle esperienze, e di rendere relativamente fungibili le presenze dei singoli salvaguardando i progetti.

Valore di non poco conto se si pensa al peso che i processi di mobilità, volontaria o accidentale, hanno nella dislocazione degli insegnanti tra le diverse unità scolastiche e sul territorio.

Senza contare l'interesse "per tutti" della possibilità di "misurare" il lavoro e di stabilire parametri di "produttività", che in questo caso significano misura della corrispondenza tra progetti e risultati che solo una dettagliata e sempre disponibile documentazione consente.

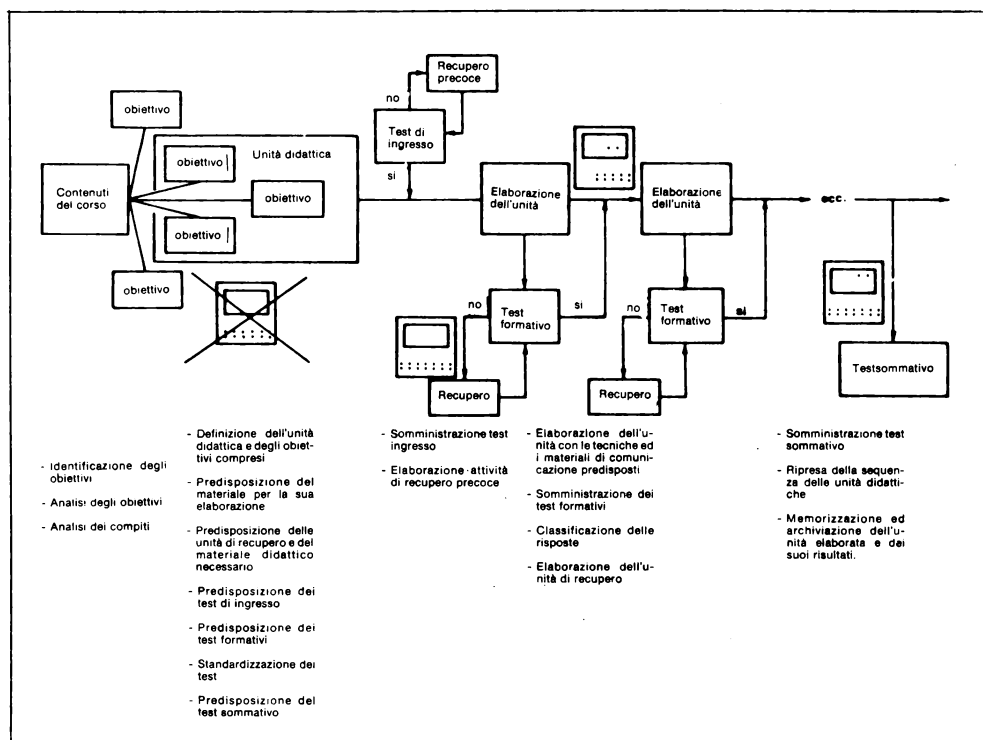


Fig. 1.1. Organizzazione del lavoro e nuova professionalità nella scuola. F. De Anna; Quaderni di battaglie del lavoro - 1982.

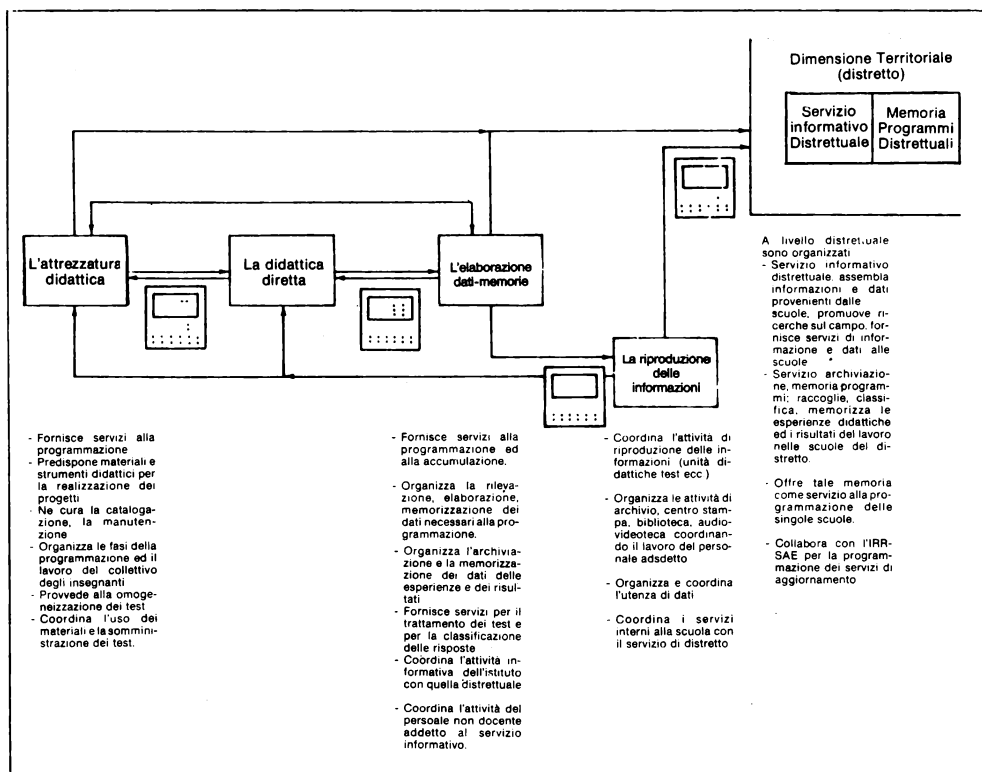


Fig. 1.2. Organizzazione del lavoro e nuova professionalità nella scuola. F. De Anna; Quaderni di battaglie del lavoro - 1982.

1.2. Unità Didattiche Software

PIERO D'ALFONSO E GIOVANNI MARTORANA

Fintanto che il software e, nel nostro caso, il software didattico sarà percepito come una costruzione complessa e monolitica l'unica strada percorribile resterà quella del puro utilizzo di programmi confezionati da informatici con tutte le riserve del caso.

Esiste però una diversa possibilità a nostro avviso assai più interessante. Merito dei linguaggi elevati, e in particolare del BASIC, che hanno suggerito ai docenti un uso più versatile del computer, pur non escludendo altre utilizzazioni tradizionali.

La possibilità che si intravede è quella di fornire ai singoli docenti gli elementi di conoscenza necessari per costruire essi stessi il proprio software al livello di complessità a loro necessario.

Questo consentirebbe a loro e ai loro alunni di risolvere problemi programmando il computer con un impegno intellettuale e temporale ragionevole.

Vogliamo qui di seguito fornire argomentazioni ed esempi che illustrino questa tesi e mostrino il tipo di lavoro che già da un paio di anni si sta sperimentando in numerose scuole e che sembra suscitare un'attenzione notevole.

È opportuno premettere che per ora ci occuperemo della programmazione dello Spectrum in linguaggio BASIC. Con questo non intendiamo riconoscere alcun primato di questo linguaggio su altri, quali LOGO, PASCAL, FORTH ecc., ci limitiamo a constatare che il primo impatto con lo Spectrum passa per il BASIC e già questo linguaggio offre buone possibilità di lavoro.

Proponiamo anzi a docenti ed esperti di sperimentare percorsi didattici analoghi e differenziati in modo da espandere la gamma di esempi e di scelte.

1.2.1. Mattoncini di software: tanti programmi con poche linee

Nella scuola, seppure con fatica, si è fatta strada tra i docenti la convinzione di dover organizzare il modello formativo per unità didattiche.

Questa frantumazione dei percorsi conoscitivi è però accettabile solo a condizione di aver previsto con anticipo la successione delle unità didattiche e aver definito una struttura che le correla; in altre parole il modello “per unità didattiche” presuppone una programmazione.

Si è voluto premettere qual è l’attuale impostazione degli studi nella scuola di base per mettere in risalto l’analogia con quanto si esporrà nel seguito.

Nel caso di unità didattiche disciplinari gli studenti sono generalmente stimolati ad assumere conoscenze esercitandosi con problemi a complessità crescente avendo come principali supporti: libro di testo, carta e matita.

I problemi affrontati e risolti possono essere visti anche alla luce di chi ricerca al loro interno quelle procedure che ricorrono secondo modelli strutturali/standard, per intenderci rappresentabili con diagrammi di flusso, al fine di far raggiungere agli studenti la concettualizzazione delle conoscenze di cui i problemi sono portatori.

I linguaggi informatici offrono in proposito la possibilità di verificare su un’ampia casistica la correttezza dell’analisi strutturale del problema, pertanto la scrittura di un programma eseguibile da un computer è la strada che dà concretezza ad una operazione intellettuale che è soprattutto di astrazione.

La percorribilità poggia però sul rapporto sforzo/benefici; qualora il linguaggio richiedesse per la stesura del programma un impegno intellettuale e temporale troppo gravoso sarebbe controproducente proporlo agli studenti.

I linguaggi elevati, e tra questi il BASIC, offrono invece la possibilità di risolvere problemi semplici con programmi di poche righe grazie a parole-chiave sintetiche e potenti.

Collezionare a fianco di problemi semplici i listati di programmi BASIC che ne forniscono la soluzione per una concreta casistica corrisponde a costruire tanti mattoncini di software la cui utilità può essere estesa oltre la diretta relazione “problem-solving/programma BASIC” per un problema specifico.

Il termine “mattoncini” è stato adoperato non a caso.

La struttura di questi miniprogrammi deve rispondere esternamente a due requisiti:

- la *compatibilità* per cui ognuno di essi deve possedere una sua funzione a cui assolve compiutamente, ma deve anche poter essere associato ad altri per assolvere ad una funzione più complessa;
- la *modularità* per cui occorre una forma esterna unificata che consenta questa composizione.

Abbiamo pensato di dare un nome e una prima forma a questi mattoncini e di mettere il tutto a disposizione di colleghi in attesa di raccogliere il prodotto del loro lavoro.

Il nome è “UDS” contrazione di Unità Didattiche Software e la forma esemplificata delle UDS è quella che presentiamo nel capitolo 2.

1.2.2. La componibilità dei programmi BASIC: il Menù e il MERGE di programmi

La scrittura di un programma in BASIC comporta che ogni istruzione sia corredata del proprio numero d'ordine dal quale dipende la posizione che dovrà occupare nella lista.

I numeri d'ordine consentiti dallo Spectrum sono in totale 10.000 (da 0 a 9999) e nulla obbliga a cominciare un programma dalla riga 10, al contrario dimostreremo che la strada da seguire è un'altra.

A questo punto occorre riflettere sulla struttura generale dei problemi.

La soluzione di un problema comporta che vengono formulate delle opzioni che sono in numero tanto maggiore quanto più il problema è complesso, ma non basta: scelta un'opzione si possono presentare un ventaglio di alternative e quindi nuove opzioni.

Ogni qual volta la soluzione del problema propone una ramificazione si potrà strutturare un Menù, cioè l'elenco delle possibili scelte: seguendo questa gerarchia si giunge a problemi semplici alla cui soluzione siamo appunto interessati.

La traduzione dei procedimenti risolutivi di problemi complessi in programmi BASIC avviene allora a due livelli:

- *strutturale*;
- *algoritmico*.

Agli aspetti *strutturali* provvedono i programmi di Menù, cioè quei componenti di programma che consentono di operare dall'esterno in base alle opzioni proposte; ogni scelta ha per conseguenza diretta l'esecuzione del sottoprogramma che le corrisponde.

I sottoprogrammi operano per *algoritmi* al fine di risolvere sottoproblemi e/o problemi parziali del problema complesso, dopo di che ritornano per ulteriori scelte al Menu di partenza.

Il modello appena descritto suggerisce di concentrare in blocchi quelle istruzioni che fanno capo ad uno stesso sottoprogramma, in altre parole basterà riservare ad ogni sottoprogramma e/o programma di Menu un intervallo di numeri d'ordine ponendo attenzione a non provocare sovrapposizioni.

Dai programmi di menu si verrà inviati ai sottoprogrammi, dopo aver compiuto una scelta, con istruzioni di salto.

Questa tecnica di affrontare problemi complessi viene ulteriormente facilitata dalla possibilità, offerta ormai anche da piccoli personal come lo Spectrum, di costruire sottoprogrammi che possono essere conservati separatamente su nastro, ma all'occorrenza richiamati e ricomposti.

Ecco che i nostri mattoncini di software possono diventare costruzioni più articolate, agganciandosi opportunamente l'uno con l'altro.

Programmi funzionanti autonomamente, fatti in tempi diversi e forse anche con scopi non tutti coincidenti, possono dunque essere ripresi e montati con un effetto costruttivo rilevante.

Basta avere l'accortezza di agire opportunamente sui numeri d'ordine associati alle istruzioni.

L'elemento che rende possibile questa aggregazione è la parola chiave MERGE.

Lasciata da parte le precisazioni di ordine tecnico, merita qui chiarire con maggiore dettaglio come opera l'istruzione composta con MERGE.

Immaginiamo di aver registrato uno dopo l'altro su cassetta una serie di sottoprogrammi che eseguono delle attività in qualche modo coerenti l'una con l'altra.

Immaginiamo che i numeri d'ordine delle istruzioni del primo siano tutti inferiori a quelli del secondo e che questi siano inferiori a quelli del successivo e così via.

È evidente che se si riuscisse a richiamarli e a montarli insieme si otterrebbe un grande

programma unico funzionante dalla prima riga del primo all'ultima riga dell'ultimo.

Se uno poi di questi sottoprogrammi fosse un Menù che definisse le attività degli altri e ad essi rinviasse, si sarebbe compiuta una costruzione unitaria e coerente di un livello più potente rispetto a quello dei singoli sottoprogrammi.

Ebbene ciò è possibile operando un caricamento da cassetta con un'istruzione MERGE.

In tal modo i sottoprogrammi e il programma di menù caricati uno dopo l'altro non si sostituiscono ogni volta nella memoria del computer, ma si accoppiano secondo la struttura prevista all'atto in cui sono stati assegnati gli intervalli di numeri d'ordine; in conclusione ne risulta un programma complesso che è composizione di sottoprogrammi.

È inoltre assai importante che il "già programmato" non vada cancellato con lo spegnimento della macchina.

La cassetta magnetica è già un buon supporto perché consente di richiamare i programmi già fatti.

Essa però non permette un MERGE di più programmi sempre sicuro e rapido.

È da poco disponibile comunque un'interfaccia molto poco costosa che consente l'affiancamento allo Spectrum di un Microdrive abbastanza capace (circa 100 K formattato), rapido e in grado di richiamare e combinare con MERGE più programmi.

1.2.3. Gli insegnanti compongono programmi complessi

L'analisi di un problema complesso e la progettazione di un sistema risolutore è compito dell'insegnante che per questo fine dispone di conoscenze culturali adeguate e competenze metodologiche.

La componibilità di programmi offerta dal BASIC può essere valorizzata se si struttura nei docenti la capacità di ridurre a problemi semplici, a schemi ed in ultima analisi, a menù, i problemi per i quali sono necessari gli algoritmi risolutivi.

La maggior parte dei programmi presentati nei capitoli che seguiranno sono stati costruiti con questa tecnica e realizzati a più mani, anche nei casi in cui portano un'unica paternità.

1.2.4 Gli alunni possono realizzare programmi semplici

Avendo a disposizione un quadro risolutore a fronte di un problema è agevole strutturare la soluzione di problemi semplici in unità didattiche per le quali lo stesso menù del programma complesso prevede sia la successione sia il grado di correlazione.

Con ciò abbiamo ritrovato proprio quella struttura di modello formativo che si fonda sulla programmazione didattica, con la differenza non trascurabile che ora la proposta di lavoro si avvale sia dei contenuti disciplinari propri del problema esaminato sia degli strumenti operativi che consentono di concretizzare le conoscenze e i metodi ad essi collegati.

I programmi semplici su cui gli studenti cimenteranno le proprie capacità risolutive si prestano ad un duplice uso: per un verso costituiranno le tessere del puzzle che il docente ha loro riservato all'atto della programmazione, per altro potranno essere mattoncini disponibili per nuove costruzioni.

Come tessere di un puzzle i programmi semplici consentono agli studenti di operare sulle relazioni col fine di far loro conoscere le modalità attraverso cui si realizzano strutture complesse.

Come mattoncini di software diventano patrimonio di tutti gli utilizzatori i quali possono operare nuove aggregazioni con il risultato di risolvere nuovi problemi complessi; in tal senso è indispensabile che anche piccoli personal come lo Spectrum abbiano a disposizione una funzione che permette di rinumerare le linee di programma per agevolare gli incastri dei mattoncini previsti dai nuovi programmi di menù.

Qui il richiamo agli insegnanti perché non solo rendano possibile la traduzione dell'algoritmo in linguaggio artificiale ma propongano la tassonomia:

- risoluzione dell'algoritmo (risolutore) di quel problema (in particolare);
- risoluzione dell'algoritmo (risolutore) di tutti i problemi come quello (in generale);
- conservazione del programma risolutivo in linguaggio artificiale su supporto magnetico (cassetta o cartridge);
- generalizzazione del problema di cui tutti i problemi come quello risolto sono un caso particolare;
- analisi di un altro caso, sua simulazione, in particolare e in generale;
- conservazione del programma risolutore in linguaggio artificiale su supporto magnetico;

attraverso la quale gli alunni verificano la possibilità di comporre (con MERGE), la risoluzione, per qualsiasi valore, di due casi particolari di un problema più generale e sono in grado di costruire, almeno mentalmente la procedura iterativa che permette di comporre più soluzioni di casi particolari.

1.2.5 Il supporto scritto delle U.D.S.

La fruibilità dei mattoncini di software non è automatica, infatti la sola lista delle istruzioni è uno strumento troppo arido ed in ogni caso incapace a stimolare nell'utente nuove curiosità e nuovi traguardi.

Si rende allora necessario un doppio commento scritto:

- didattico;
- algoritmico.

Il primo deve precisare le circostanze operative che hanno favorito la costruzione e l'uso del programma; il secondo deve fornire con l'algoritmo le modalità risolutive e i limiti di impiego.

Senza questo supporto scritto che riteniamo fondamentale e che permette di discutere sull'uso del software messo a disposizione poco può essere fatto a scuola.

Sarebbe come fornire agli alunni la proprietà di una figura geometrica senza indicarne le conseguenze e l'utilità concreta.

Per concludere, dunque, la strategia che porta al software didattico anche complesso (di cui parleremo nel capitolo successivo), passa attraverso strutture di software elementari facilmente componibili e modulari.

Il software, anche complesso purché facilmente scomponibile, è facilmente comprensibile a patto che esista un valido supporto scritto che lo illustri e lo renda modificabile.

Nel capitolo che segue forniamo alcuni esempi di UDS realizzate da insegnanti di scuole medie.

CAPITOLO II

Esempi U.D.S.

2.1 Scomposizione in fattori

Servizio U.D.S.

CODICE	MAT/1		
AUTORE	LUISA TINELLI	ANNO 1984	
SCUOLA	MEDIA STATALE "CASA DEL SOLE" - MILANO		
TITOLO	Scomposizione in fattori primi		

RIASSUNTO

A partire da una prima conoscenza della scomposizione in fattori primi lo studente è condotto a trasferire operazioni che sa eseguire mnemonicamente in un preciso algoritmo e dimostra di padroneggiarlo insegnando al computer a rieseguirlo

Curatori: PIERO D'ALFONSO - GIOVANNI MARTORANA

Elementare	<input type="checkbox"/>	Alfabetizzazione informatica	<input type="checkbox"/>
Media	<input checked="" type="checkbox"/>	linguistica	<input type="checkbox"/>
Media superiore	<input type="checkbox"/>	scienze / matematica	<input checked="" type="checkbox"/>
		tecnica	<input type="checkbox"/>
		grafica	<input type="checkbox"/>

BASIC	<input checked="" type="checkbox"/>	LOGO	<input type="checkbox"/>	PASCAL	<input type="checkbox"/>
-------	-------------------------------------	------	--------------------------	--------	--------------------------

iterazione FOR/NEXT	<input checked="" type="checkbox"/>	(sclto per BASIC)	
scelta condizionata IF/THEN	<input checked="" type="checkbox"/>		
matrici di memoria DIM	<input type="checkbox"/>		
READ/DATA	<input type="checkbox"/>		
DATA su FILE	<input type="checkbox"/>		
soubroutine GOSUB/RETURN	<input type="checkbox"/>		
menù	<input type="checkbox"/>		
memoria di schermo	<input type="checkbox"/>		
word processing	<input type="checkbox"/>		
collegamenti a rete	<input type="checkbox"/>		
.....			

SINCLAIR ZX 81	<input type="checkbox"/>		
SINCLAIR SPECTRUM	<input checked="" type="checkbox"/>	MINIDRIVE	<input type="checkbox"/>
SINCLAIR QL	<input type="checkbox"/>		
COMMODORE VIC 20	<input type="checkbox"/>	DRIVE PER FLOPPY	<input type="checkbox"/>
COMMODORE C 64	<input type="checkbox"/>	DRIVE PER FLOPPY	<input type="checkbox"/>
M 10 OLIVETTI	<input type="checkbox"/>		
M 20 OLIVETTI	<input type="checkbox"/>	DRIVE	<input type="checkbox"/>
SHARP MZ 700	<input type="checkbox"/>	DRIVE	<input type="checkbox"/>
APPLE II	<input type="checkbox"/>	DRIVE PER FLOPPY	<input type="checkbox"/>
I.B.M. PC	<input type="checkbox"/>	DRIVE	<input type="checkbox"/>

Progetto "Banca delle Unità Ditattiche"

Obiettivi che gli alunni dovranno raggiungere:

- 1) conoscere e utilizzare il metodo di scomposizione in fattori primi attraverso l'analisi dell'algoritmo;
- 2) tradurre l'algoritmo di scomposizione in Programma BASIC (eventualmente modificato e semplificato rispetto al programma qui presentato);
- 3) utilizzare il programma per verificare la correttezza della scomposizione eseguita con carta e penna.

Prerequisiti richiesti per poter raggiungere gli obiettivi:

per l'obiettivo 1) l'alunno: — conosce e sa usare i simboli dei diagrammi di flusso
— conosce il significato di variabile e distingue il nome della variabile dal suo contenuto
— conosce il significato dei termini: divisore, quoziente, differenza, fattore, numero primo

per l'obiettivo 2) l'alunno: — conosce le due strutture elementari di programmazione: la struttura condizionale (IF... THEN) e la struttura iterativa (FOR... NEXT)
— conosce e sa usare le parole-chiave: REM, PRINT, INPUT, LET e le funzioni: AT, TAB

per l'obiettivo 3) l'alunno: — sa scomporre un numero in fattori primi (con carta e penna)
— sa usare la tastiera del computer per inserire dati

Il raggiungimento dell'obiettivo 3) è indipendente da qualsiasi conoscenza di programmazione e di linguaggio BASIC; il programma descritto nella UDS viene in questo caso utilizzato come verifica della correttezza del procedimento eseguito dall'alunno con carta e penna.

2.1.1 Introduzione

Questa UDS si rivolge ai docenti di scuola media (anche di scuola superiore qualora l'obiettivo sia quello di una riflessione sulla logica delle procedure che gli studenti applicano in modo automatico).

L'UDS vuole essere un suggerimento di come sia possibile risolvere un problema, in questo caso la scomposizione in fattori primi, rappresentando la procedura risolutiva mediante un grafo, il diagramma di flusso, che esprime le singole operazioni da compiere, le condizioni da considerare e le verifiche da eseguire per procedere in modo ordinato e corretto.

Nel corso dell'unità si affronta il problema della formulazione delle istruzioni che devono essere espresse in modo da essere comprese senza ambiguità. L'UDS può essere utilizzata in classe dopo una puntuale verifica del possesso da parte degli alunni dei prerequisiti richiesti relativi agli obiettivi che il docente intende raggiungere.

N.B. La numerazione delle istruzioni è stata scelta per rendere compatibile questa U.D.S. con altre cui può essere accostata.

Questo programma può essere utilizzato dallo studente in modi diversi:

- A) per verificare se la scomposizione in fattori primi da lui eseguita con carta e penna è corretta: questa fase non richiede alcuna conoscenza del linguaggio BASIC, sono sufficienti alcune informazioni sull'uso della tastiera;
- B) per ricostruire il programma descrivendo l'algoritmo risolutivo e rappresentandolo mediante un diagramma di flusso;
- C) per analizzare il programma stesso ed apportarvi eventuali personali modifiche.

Nei casi B e C lo studente deve avere precedenti conoscenze relative alle 3 strutture elementari di programmazione attraverso cui è possibile realizzare qualsiasi algoritmo:

— La struttura condizionale:

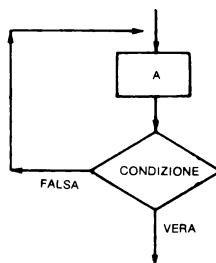
condizione nuova istruzione A
 IF THEN
 Istruzione B



— La struttura iterativa

variabile num
 FOR = n° 1 TO n° 2
 istruzioni

 variabile num
 NEXT

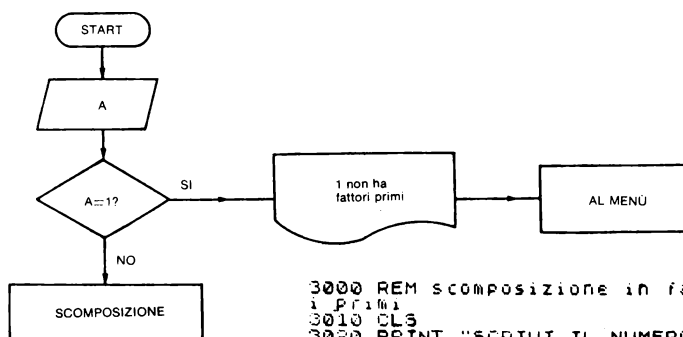


2.1.2 Procedura risolutiva

Il programma è suddiviso in due parti:

- 1 - inserimento del numero da scomporre;
- 2 - procedura per la scomposizione e la stampa;

1 - La procedura seguita per l'inserimento del numero - evitando automaticamente 1 - è descritta dal seguente Flow Chart cui corrisponde il relativo Programma BASIC.



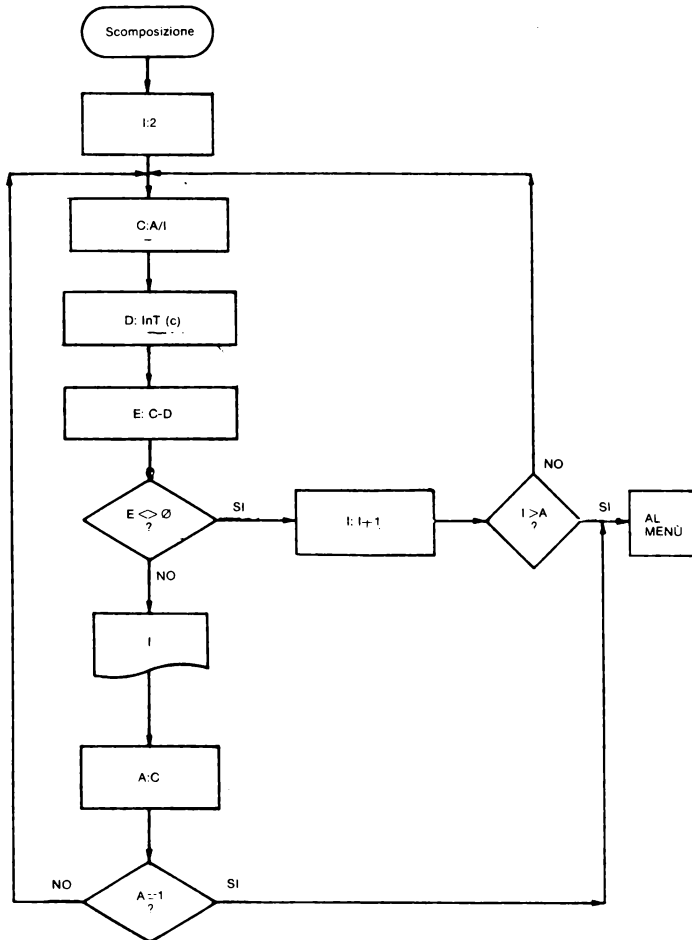
```

3000 REM scomposizione in fattori
i=1
3010 CLS
3020 PRINT "SCRIVI IL NUMERO CHE
VUOI", "SCOMPORRE:";
3030 INPUT A
3040 PRINT A
3050 IF A=1 THEN GO TO 3200
3200 PRINT AT 5,0;"IL NUMERO 1 N
ON HA FATTORI PRIMI"
3210 RETURN
  
```

Nell'indirizzo di memoria A si inserisce il numero da scomporre linea 3030. Da questa linea alla 3080 vi sono semplici istruzioni di stampa. Se la condizione espressa dalla linea 3080 fosse vera, verrebbe eseguita l'istruzione di salto condizionato che, evitando la procedura di scomposizione,

porta direttamente alla linea 3200 in quanto il numero 1 non ha fattori primi.

2 - Il Flow Chart relativo alla procedura di scomposizione in fattori primi è il seguente:



Elenco e descrizione delle variabili utilizzate:

A rappresenta un indirizzo di memoria in cui è stato precedentemente inserito il numero da scomporre. In A possono ovviamente essere inseriti solo numeri interi.

I rappresenta la variabile del ciclo di iterazione che assume 2 come valore iniziale e A come valore finale.

C contiene il quoziente della divisione tra A e I

D contiene la parte intera del numero memorizzato in C

E contiene la differenza tra C e D (cioè tra il numero e la sua parte intera)

Con I si cercano i fattori (infatti I conta: 2, 3, 4, 5 ... fino ad A);

con C si divide ogni volta il numero di cui si fa la scomposizione (contenuto in A) per i fattori che si cercano (contenuti in I).

Se il quoziente della divisione (C) è intero, I è uno dei fattori cercati. Per saperlo basta controllare la differenza tra C e D (parte intera di C). In questo caso il quoziente C moltiplicato per I dà esattamente A.

I va stampato e C va a sostituire il numero originario nell'indirizzo di memoria A.

Proseguendo, l'algoritmo sottopone alla stessa prova il nuovo numero. I riparte da 2 e non si modificherà fin quando non troverà un valore in E (C-D) diverso da 0 (in questo caso I non può essere un fattore primo di A e pertanto I viene incrementato di una unità).

È chiaro quindi che mai verrà riconosciuto come fattore primo un numero composto (4, 6, 8...) infatti se A fosse 4 l'avvicendamento dei valori nella tabella di memoria sarebbe il seguente:

A	4	2	1
I	2	2	
C	4:2=2	2:2=1	
D	2	1	
E	2-2=0	0	

Infatti $4 = 2 \times 2$

Prima di tradurre l'algoritmo in programma BASIC seguiamo lo svolgimento della procedura con un esempio concreto osservando il comportamento delle variabili.

- A 20 in A viene inserito 20
 I 2 I assume il valore 2
 C 10 C assume valore 10 (in questo caso $A/I = 20/2$)
 D 10 D assume valore 10 (C è uguale a D perché il
 quoziente di A/I è un numero intero),
 pertanto
 E 0 E assume valore 0 (in questo caso $C-D=10-10$)

Poiché E è uguale a zero il contenuto di I (cioè 2) è un fattore primo di 20 e quindi viene stampato.

Occorre ora immettere nella memoria di A il valore contenuto nella memoria C: il 20 viene cancellato e sostituito dal 10.

Il valore di I è ancora 2 e tale valore non verrà modificato fino a che il quoziente tra A e I sarà un numero intero.

Tabella delle memorie completa:

A	20	10	5	5	5	5	1
I	2	2	2	3	4	5	
C	10	5	2,5	1,6	1,25	1	
D	10	5	2	1	1	1	
E	0	0	0,5	0,6	0,25	0	

Osservando la tabella delle memorie rispondi:

- 1) Perché la variabile I non assume mai il valore 1?
(R. 1 non è un fattore primo)!
- 2) Quale delle variabili A, I, C, D, E contiene i fattori primi? (R.I)
- 3) Quando occorre inserire nella memoria A il contenuto della memoria C (R. quando il quoziente della divisione A/I è un n° intero)
- 4) Quando la variabile di ciclo I viene incrementata di una unità? (R. quando $E < > 0$)

Programma Basic relativo alla scomposizione in fattori primi.

```
3050>PRINT "SCOMPOSIZIONE IN FAT
TORI PRIMI DEL NUMERO: ";A
3060 PRINT
3080 IF A=1 THEN GO TO 3200
3085 PRINT AT 4,14-LEN STR$ (A);
A=I
3090 FOR I=2 TO A
3100 LET C=A/I
3110 LET D=INT (C)
3120 LET E=C-D
3130 IF E<>0 THEN GO TO 3180
3150 PRINT I
3160 LET A=C
3165 PRINT TAB 14-LEN STR$ (A);A
A=I
3168 IF A=1 THEN GO TO 3210
3170 GO TO 3100
3180 NEXT I
3200 PRINT AT 5,0;"IL NUMERO 1 N
ON HA FATTORI PRIMI"
3210 RETURN
```

Analizziamo le istruzioni di stampa (linee 3085 - 3150 - 3165) per la scrittura in colonne dei numeri della scomposizione.

```
3085 PRINT AT 4,14-LEN STR$ (A);
A
```

La funzione LEN stringa dà la lunghezza in caratteri della stringa:

Esempio: LEN "CIAO" dà 4 (ciao è formato da 4 caratteri)

La funzione STR \$ (numero) dà la stringa di caratteri che corrisponde alla cifre del numero.

Nel nostro esempio:

```
LEN STR$ (20)
```

Ciò significa che il numero 20 verrà stampato alla riga 4 a partire dalla colonna 12:

```
3150>PRINT I
```

Il valore di I, cioè il fattore primo, viene stampato sulla stessa riga, a partire dalla colonna 16 (ciò è dovuto alla virgola che concludeva la linea 3085):

```
3165>PRINT TAB 14-LEN STR$ (A);A
```

La linea 3165 ricalca sostanzialmente la linea 3085 (al posto della funzione AT si trova ora la funzione TAB in quanto interessa solo incolonnare correttamente andando a capo ogni volta)

Osservando il programma rispondi:

- 1) Modifica le istruzioni di stampa in modo da ottenere sulla pagina video la scomposizione scritta diversamente. (esempio: $20=2 \times 2 \times 5$)
- 2) Se il numero da scomporre fosse 3000 in quale colonna inizierebbe la scrittura del numero? (osserva la linea 3085) (R. 10).

Si riporta il programma completo:

```
3000 REM scomposizione in fattor
i primi
3010 CLS
```

```

3020 PRINT "SCRIVI IL NUMERO CHE
VUOI"; "SCOMPORRE: ";
3030 INPUT A
3040 CLS
3050 PRINT "SCOMPOSIZIONE IN FAT
TORI PRIMI DEL NUMERO: ";A
3060 PRINT
3080 IF A=1 THEN GO TO 3200
3085 PRINT AT 4,14-LEN STR$ (A);
A; " "
3090 FOR I=2 TO A
3100 LET C=A/I
3110 LET D=INT (C)
3120 LET E=C-D
3130 IF E<>0 THEN GO TO 3180
3150 PRINT I
3160 LET A=C
3165 PRINT TAB 14-LEN STR$ (A);A
" "
3168 IF A=1 THEN GO TO 3210
3170 GO TO 3100
3180 NEXT I
3200 PRINT AT 5,0;"IL NUMERO 1 N
ON HA FATTORI PRIMI"
3210 RETURN

```

2.1.3 Osservazioni

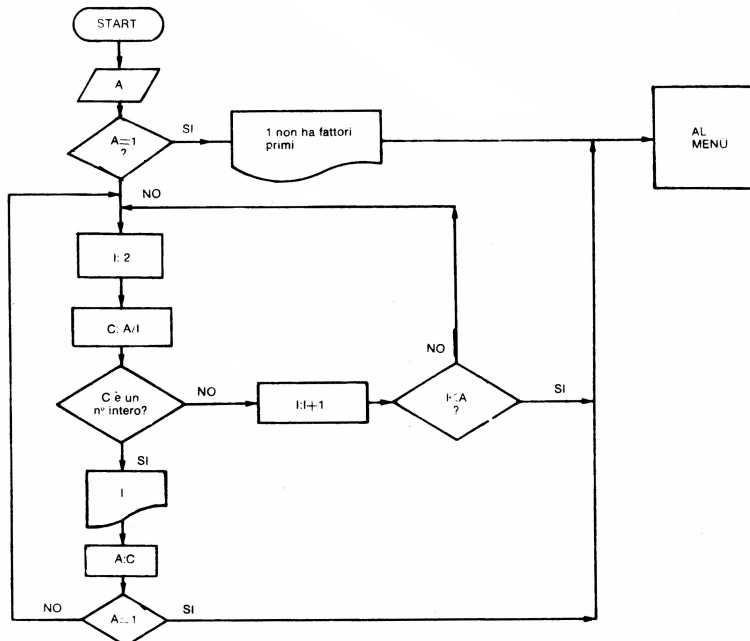
L'algoritmo presentato nell'UDS è costruito pensando all'esecutore-elaboratore, cioè ad un esecutore a cui occorre specificare tutti i singoli passaggi affinché sia in grado di compiere una determinata operazione.

Diverso è l'esecutore-studente che già possedendo nella sua memoria una quantità di conoscenze necessita un minor numero di informazioni per procedere correttamente.

È quindi importante nella costruzione di un algoritmo sapere a che tipo di esecutore ci si rivolge.

In un primo momento sarà opportuno costruire con la classe un algoritmo semplificato a misura di alunno in modo che siano gli alunni stessi in grado di capire e quindi di prevedere le diverse necessità di un esecutore automatico.

L'algoritmo proposto è il seguente:



Le variabili utilizzate corrispondono esattamente a quelle descritte in precedenza (ad esclusione delle variabili D ed E che non sono state utilizzate).

Questo algoritmo è più semplice perché qualsiasi studente sa riconoscere immediatamente se un quoziente è intero o decimale (cioè sa rispondere alla domanda: C è un numero intero?), ma il computer no!

Solo in un secondo momento quando gli alunni avranno capito il procedimento e lo sapranno utilizzare con sicurezza si dovrà porre il problema dell'esecutore automatico che ha bisogno di "imparare" come si riconosce un numero intero da un numero decimale.

Il metodo utilizzato nell'algoritmo della pagina precedente si basa sul fatto che se si sottrae da un numero la sua parte intera la differenza è uguale a zero quando il numero è intero, è diversa da zero se il numero è decimale.

2.2 Trasformazione di un numero da base 2 a base 10

Servizio U.D.S.

CODICE MAT 8/a

AUTORE PAOLA CATALANI

SCUOLA MEDIA STATALE "DI VITTORIO" - Pioltello (Mi)

TITOLO "Punti sul piano cartesiano" (4 Quadranti)

RIASSUNTO

Si costruisce un piano cartesiano di quattro quadranti in cui i punti vengono inseriti con le coordinate relative al sistema presente sullo schermo anziché quello automatico dell'elaboratore.

Curatori: PIERO D'ALFONSO - GIOVANNI MARTORANA

Elementare	<input type="checkbox"/>	Alfabetizzazione informatica	<input type="checkbox"/>
Media	<input checked="" type="checkbox"/>	linguistica	<input type="checkbox"/>
Media superiore	<input type="checkbox"/>	scienze / matematica	<input checked="" type="checkbox"/>
		tecnica	<input type="checkbox"/>
		grafica	<input type="checkbox"/>
BASIC	<input checked="" type="checkbox"/>	LOGO	<input type="checkbox"/>
		PASCAL	<input type="checkbox"/>
iterazione FOR/NEXT	<input checked="" type="checkbox"/>	(solo per BASIC)	
scelta condizionata IF/THEN	<input checked="" type="checkbox"/>		
matrici di memoria DIM	<input type="checkbox"/>		
READ/DATA			
DATA su FILE	<input type="checkbox"/>		
soubroutine GOSUB/RETURN	<input type="checkbox"/>		
menù	<input type="checkbox"/>		
memoria di schermo	<input type="checkbox"/>		
word processing	<input type="checkbox"/>		
collegamenti a rete	<input type="checkbox"/>		
.....			
SINCLAIR ZX 81	<input type="checkbox"/>		
SINCLAIR SPECTRUM	<input checked="" type="checkbox"/>	MINIDRIVE	<input checked="" type="checkbox"/>
SINCLAIR QL	<input type="checkbox"/>		
COMMODORE VIC 20	<input type="checkbox"/>	DRIVE PER FLOPPY	<input type="checkbox"/>
COMMODORE C 64	<input type="checkbox"/>	DRIVE PER FLOPPY	<input type="checkbox"/>

M 10 OLIVETTI	<input type="checkbox"/>		
M 20 OLIVETTI	<input type="checkbox"/>	DRIVE	<input type="checkbox"/>
SHARP MZ 700	<input type="checkbox"/>	DRIVE	<input type="checkbox"/>
APPLE II	<input type="checkbox"/>	DRIVE PER FLOPPY	<input type="checkbox"/>
I.B.M. PC	<input type="checkbox"/>	DRIVE	<input type="checkbox"/>

Progetto "Banca delle Unità Didattiche"

N.B. La numerazione delle istruzioni è stata scelta per rendere compatibile questa UDS con altre cui può essere associata.

Questo programma può essere utilizzato dallo studente in modi diversi:

- A) per verificare se la trasformazione di un numero da base 2 a base 10 da lui eseguita con carta e penna è corretta (questa fase non richiede alcuna conoscenza del linguaggio BASIC, sono sufficienti alcune informazioni sull'uso della tastiera)
- B) per costruire il programma descrivendo l'algoritmo risolutivo e rappresentandolo mediante un diagramma di flusso.
- C) per analizzare il programma stesso ed apportarvi eventuali personali modifiche.

Nei casi B e C lo studente deve già possedere conoscenze relative alle due strutture elementari di programmazione attraverso cui è possibile realizzare qualsiasi algoritmo:

- la struttura condizionale (IF... THEN...)
- la struttura iterativa (FOR... NEXT)

In questa UDS si utilizzano le funzioni LEN e VAL che non vengono spiegate in quanto oggetto di un'altra UDS.

Il sistema binario utilizza solo due simboli 0 e 1; l'uso di questo sistema fu suggerito da Gottfried Leibniz (1646-1716), ma solo recentemente se ne è capita l'utilità perché i computer eseguono i calcoli proprio utilizzando questo sistema.

Ecco come si può contare avendo solo due cifre disponibili:

base 10	base 2
0	0
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000

Dalla tabella si osserva che la cifra 1 assume valori diversi a secondo della posizione che occupa (le posizioni si numerano da destra a sinistra):

- in I posizione vale 1 (2^0)
- in II posizione vale 2 (2^1)
- in III posizione vale 4 (2^2)
- in IV posizione vale 8 (2^3)

Il sistema binario è quindi come quello decimale un sistema posizionale.

Come è allora possibile trasformare un numero binario in decimale?

- La prima operazione da compiere è quella di associare ad ogni cifra il valore della corrispondente potenza di 2.

Ad esempio dato il numero binario 1101 possiamo rappresentare l'associazione di ogni cifra con la corrispondente potenza di 2 e quindi con il valore decimale come mostra la tabella seguente:

1	1	0	1
2^3	2^2	2^1	2^0
8	4	2	1

— La seconda operazione è quella di determinare il valore decimale di ogni cifra binaria; ciò si ottiene moltiplicando ogni cifra per il corrispondente valore della potenza di 2:

$$\begin{aligned} 1 \times 1 &= 1 \\ 0 \times 2 &= 0 \\ 1 \times 4 &= 4 \\ 1 \times 8 &= 8 \end{aligned}$$

— Finalmente, sommando i valori ottenuti si ottiene il numero a base 10:

$$1 + 0 + 4 + 8 = 13$$

$$\text{Perciò: } 1101_2 = 13_{10}$$

Esercizio:

Dato il numero 111010_2 trova il corrispondente a base 10.

Completa la tabella:

1	1	1	0	1	0
...	2^1	2^0
...	2	1

Esegui le moltiplicazioni e somma infine i valori trovati:

$$\begin{aligned} 1 \times \dots &= \dots \\ 1 \times \dots &= \dots \\ 1 \times \dots &= \dots \\ 0 \times \dots &= \dots \\ 1 \times \dots &= \dots \\ 0 \times \dots &= \dots \end{aligned}$$

$$\text{Perciò } 111010_2 = \dots$$

Questo procedimento che abbiamo eseguito con carta e penna vogliamo ora tradurlo in una procedura che sia eseguibile da un esecutore automatico come il computer.

La procedura e di conseguenza il programma devono prevedere due momenti successivi:

1. L'inserimento del numero binario
2. L'algoritmo che trasforma il numero da base 2 a base 10

1. Per inserire il numero binario occorre tenere presente che per risolvere il nostro problema noi dobbiamo essere in grado di prevalere ogni singola cifra del numero da un definito spazio di memoria; per far ciò potremo utilizzare una variabile numerica?

Ricordiamo il diverso comportamento delle variabili numeriche e alfanumeriche, provando i seguenti brevi programmi.

```
10 LET A=1101
20 PRINT A
25 REM tentativo di stampare
   la prima cifra
30 PRINT A(1)
```

Premi RUN: in alto sullo schermo comparirà

1101

e in basso la scritta di errore

2 Variabile not found, 30:1.

L'istruzione 30 non è efficace: inserendo il numero nella variabile numerica A non è possibile richiamare le singole cifre, perché il numero è memorizzato nel suo complesso in un'unica locazione di memoria.

Prova ora questo programma:

```
10 LET A$="1101"
20 PRINT A$
30 PRINT A$(1)
40 PRINT A$(2)
50 PRINT A$(3)
60 PRINT A$(4)
```

Sul video comparirà

```
1101
1
1
0
1
```

Ogni cifra si trova ora in una posizione precisa e può essere richiamata con facilità.

Per conoscere quante cifre compongono il numero, si utilizza la funzione BASIC LEN che dice di quanti caratteri è composta una stringa.

Prova ora questo programma:

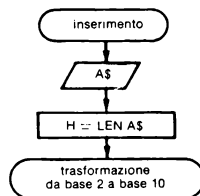
```
10 LET A$="1101"
20 PRINT A$
30 LET H=LEN A$
40 PRINT H
```

Sul video comparirà:

```
1101
4
```

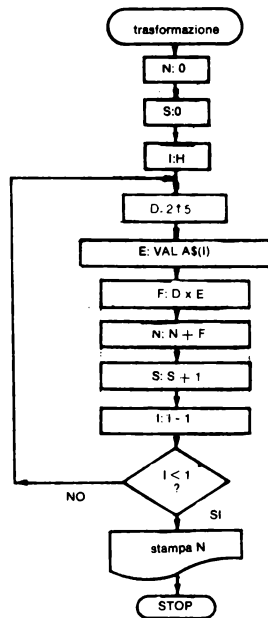
Nella variabile H è quindi inserito il numero dei caratteri del dato memorizzato in A\$.

Ecco il diagramma di flusso e il programma relativi all'inserimento del numero binario:



```
1000 REM da base 2 a base 10
1005 CLS
1015 INPUT A$
1020 LET H=LEN A$
```

2. L'algoritmo per la trasformazione del numero da base 2 a base 10 può essere rappresentato con il seguente diagramma di flusso:



Elenco e descrizione delle variabili utilizzate.

- S è la variabile destinata agli esponenti di 2. Assumerà valori compresi tra 0 e H - 1, infatti ogni cifra binaria dovrà essere messa in corrispondenza con la potenza di 2 relativa alla posizione della cifra stessa. Per il numero 1101 le corrispondenze saranno:

1	1	0	1
2^3	2^2	2^1	2^0

S quindi varia, attraverso incrementi unitari, da 0 a 3 e poiché H in questo esempio avrà valore 4 il campo di variabilità di S è proprio tra 0 e H - 1.

- I è la variabile del ciclo di iterazione che permette di prendere in considerazione una cifra per volta del numero memorizzato in A\$.

Possiamo immaginare la memoria A\$ dove è contenuto il numero 1101 così strutturata:

A\$(1)	A\$(2)	A\$(3)	A\$(4)
1	1	0	1

Osserviamo che la cifra che si trova in I° posizione (quella che dovrà essere messa in corrispondenza con 2^0) si trova A\$(4), la cifra che si trova in II° posizione (che dovrà essere messa in corrispondenza con 2^1 si trova in A\$(3) ...), perciò la variabile I dovrà avere inizialmente valore 4 (cioè il contenuto di H) e con decrementi unitari raggiungere, alla fine del ciclo, valore 1. Pertanto I varierà da H a 1 STEP - 1 (STEP - 1 indica il decremento unitario).

- D contiene il valore della potenza di 2. Riferendoci allo stesso esempio:

se S = 0 allora D = 1 perché $2^0 = 1$
 se S = 1 allora D = 2 perché $2^1 = 2$
 se S = 2 allora D = 4 perché $2^2 = 4$
 se S = 3 allora D = 8 perché $2^3 = 8$

- E contiene il valore della cifra nella posizione voluta
 se $E = \text{VAL } A\$(4)$ allora E avrà valore 1
 se $E = \text{VAL } A\$(3)$ allora E avrà valore 0
 se $E = \text{VAL } A\$(2)$ allora E avrà valore 1
 se $E = \text{VAL } A\$(1)$ allora E avrà valore 1

Si utilizza la funzione VAL per poter eseguire calcoli con numeri memorizzati in variabili alfanumeriche.

Verifica sul computer la funzione di VAL scrivendo il programma:

```
10 LET A$="1101"
20 LET E=VAL A$(1)
30 PRINT VAL A$(1)
40 PRINT E*8
```

Sul video comparirà

$\frac{1}{8}$

Prova ora a richiedere in diretta:

```
PRINT A$(1)*8
```

Che cosa succede?

- F contiene il valore decimale della cifra binaria ottenuto moltiplicando ogni cifra per la corrispondente potenza di 2.

$F = D \times E$

Nel nostro esempio F assumerà i valori:

$F = 1$ quando $F = 2^0 \times 1$

$F = 0$ quando $F = 2^1 \times 0$

$F = 4$ quando $F = 2^2 \times 1$

$F = 8$ quando $F = 2^3 \times 1$

Proprio nella memoria F sono contenuti i valori che devono essere sommati per ottenere il numero in base 10.

Affinché la macchina esegua automaticamente questa somma occorre utilizzare una nuova variabile che chiameremo N.

- N è un sommatore avrà all'inizio valore 0 in quanto non ha ancora ricevuto l'ingresso di nessun numero.

Successivamente N assumerà valore $N + F$ e passerà attraverso i seguenti stati:

$N = 1$ quando $F = 1$

$N = 1$ quando $F = 0$

$N = 5$ quando $F = 4$

$N = 13$ quando $F = 8$

A trasformazione conclusa in N è contenuto il numero trasformato da base 2 a base 10 e potrà quindi essere stampato.

Il programma relativo alla trasformazione del numero binario in decimale è il seguente:

```
1040>REM trasformazione del nume
ro da base 2 a base 10
1041 CLS
1045 PRINT "NUMERO A BASE 2:";AT
0,32-H;A$
1055 LET N=0
1056 LET S=0
1060 FOR I=H TO 1 STEP -1
1065 LET O=2^I
1070 LET E=VAL A$(I)
1075 LET F=O*E
1085 LET N=N+F
1090 LET S=S+1
1100 NEXT I
1110 PRINT AT 7+I,0;"NUMERO IN B
ASE 10:";PRINT AT 7+I,32-LEN S
TR$ (N);N
1115 RETURN
```

Prova ora a trasformare il numero binario 11001 in decimale seguendo l'algoritmo e compilando la tabella delle memorie:

A\$	11001					
H	5					
N	0					
S	0	1	2	3	4	5
I	5	4	3	2	1	0
D	1					
E	1					
F	1					

Che numero hai ottenuto?

Il programma completo è il seguente:

```

1000 REM da base 2 a base 10
1005 CLS
1010 PRINT "SCRIVI IL NUMERO A B
ASE 2"
1015 INPUT A$
1016 PRINT A$
1020 LET H=LEN A$
1025 FOR L=1 TO H: IF A$(L)<>"0"
AND A$(L)<>"1" THEN GO TO 1300
1030 NEXT L
1040 REM trasformazione del nume
ro da base 2 a base 10
1041 CLS
1045 PRINT "NUMERO A BASE 2:";AT
0,32-H;A$
1055 LET N=0
1060 LET S=0
1065 FOR I=H TO 1 STEP -1
1070 LET E=VAL A$(I)
1075 LET F=D+E
1085 LET N=N+F
1090 LET S=S+1
1100 NEXT I
1110 PRINT AT 7+I,0;"NUMERO IN B
ASE 10:";PRINT AT 7+I,32-LEN S
TR$(N);N
1115 RETURN
1300 PRINT AT 5,0;"HAI SBAGLIATO!
":AT 7,0;"UN NUMERO A BASE 2 CON
TIENE SOLO LE CIFRE 0 E 1";AT 10
,0;"PER CONTINUARE E CORREGGERE
PREMI UN TASTO QUALSIASI"
1305 IF INKEY$<>"" THEN GO TO 13
05
1310 IF INKEY$="" THEN GO TO 131
0
1315 IF INKEY$<>"" THEN GO TO 10
00

```

Osserviamo le linee 1025 1030 1300 1305 1310 1315

Queste linee non riguardano direttamente il problema della trasformazione di un numero da base 2 a base 10 ma sono ugualmente molto importanti perché impediscono alla macchina di proseguire nella trasformazione quando il numero inserito non è a base 2, cioè contiene cifre diverse da 0 e 1.

Le linee 1025 e 1030 rappresentano un'iterazione che scorrendo cifra per cifra eseguono un confronto con le cifre 0 e 1; se sono presenti cifre diverse rimandano alla linea 1300 altrimenti il programma prosegue con la trasformazione del numero.

La linea 1300 segnala errore.

Le linee 1305 e 1310 bloccano la macchina fino a che un tasto viene premuto ed allora la linea 1315 rimanda all'inizio del programma stesso.

2.3 Rappresentazione di punti sul piano cartesiano (4 Quadranti)

Servizio U.D.S.

CODICE MAT	8/a
AUTORE	PAOLA CATALANI
SCUOLA	MEDIA STATALE "DI VITTORIO" - Pioltello (Mi)
TITOLO	"Punti sul piano cartesiano" (4 Quadranti)

RIASSUNTO

Si costruisce un piano cartesiano di quattro quadranti in cui i punti vengono inseriti con le coordinate relative al sistema presente sullo schermo anziché quello automatico dell'elaboratore.

Curatori: PIERO D'ALFONSO - GIOVANNI MARTORANA

Elementare	<input type="checkbox"/>	Alfabetizzazione informatica	<input type="checkbox"/>
Media	<input checked="" type="checkbox"/>	linguistica	<input type="checkbox"/>
Media superiore	<input type="checkbox"/>	scienze / matematica	<input checked="" type="checkbox"/>
		tecnica	<input type="checkbox"/>
		grafica	<input type="checkbox"/>
BASIC	<input checked="" type="checkbox"/>	LOGO	<input type="checkbox"/>
		PASCAL	<input type="checkbox"/>
iterazione FOR/NEXT	<input checked="" type="checkbox"/>	(solo per BASIC)	
scelta condizionata IF/THEN	<input checked="" type="checkbox"/>		
matrici di memoria DIM	<input type="checkbox"/>		
READ/DATA			
DATA su FILE	<input type="checkbox"/>		
soubroutine GOSUB/RETURN	<input type="checkbox"/>		
menù	<input type="checkbox"/>		
memoria di schermo	<input type="checkbox"/>		
word processing	<input type="checkbox"/>		
collegamenti a rete	<input type="checkbox"/>		
.....			
SINCLAIR ZX 81	<input type="checkbox"/>		
SINCLAIR SPECTRUM	<input checked="" type="checkbox"/>	MINIDRIVE	<input checked="" type="checkbox"/>
SINCLAIR QL	<input type="checkbox"/>		
COMMODORE VIC 20	<input type="checkbox"/>	DRIVE PER FLOPPY	<input type="checkbox"/>
COMMODORE C 64	<input type="checkbox"/>	DRIVE PER FLOPPY	<input type="checkbox"/>
M 10 OLIVETTI	<input type="checkbox"/>		
M 20 OLIVETTI	<input type="checkbox"/>	DRIVE	<input type="checkbox"/>
SHARP MZ 700	<input type="checkbox"/>	DRIVE	<input type="checkbox"/>
APPLE II	<input type="checkbox"/>	DRIVE PER FLOPPY	<input type="checkbox"/>
I.B.M. PC	<input type="checkbox"/>	DRIVE	<input type="checkbox"/>

Progetto "Banca delle Unità Didattiche"

- Il programma è così articolato:
- Costruzione del piano cartesiano di riferimento di 4 quadranti
 - Inserimento punti

2.3.1 Costruzione assi

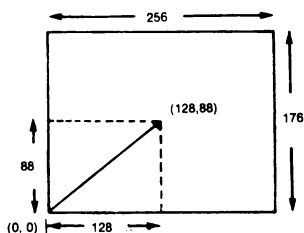
La pagina video in configurazione PLOT permette di sistemare punti con ascissa (x) variabile da 0 a 255 e ordinata (y) compresa tra i valori 0 e 175.

La parola chiave PLOT seguita dai valori x, y delle coordinate dei punti non accetta valori negativi né decimali, ma unicamente interi e positivi compresi negli intervalli specificati in precedenza.

In questo modo il video si identifica con il I° quadrante di un piano cartesiano di cui non è disegnato il sistema di riferimento, nel quale: $0 \leq x \leq 255$ e $0 \leq y \leq 175$.

In alcune occasioni può rivelarsi utile utilizzare l'intero piano cartesiano in un sistema di riferimento chiaro e leggibile, in cui cioè gli assi siano graduati in base alle necessità dell'utente.

Per ottenere un piano cartesiano (4 quadranti) sul video si deve trasportare l'origine O (0,0) al centro dello schermo:



operare cioè una traslazione del punto O dalle coordinate (0,0) a (256/2, 176/2) cioè (128,88).

I valori massimali di x e y sono stati incrementati di un'unità in quanto è necessario considerare anche lo 0 nel totale delle possibilità a disposizione per i valori delle coordinate dei punti

```
10>REM **costruzione assi**
30 PLOT 128,0: DRAW 0,175
35 PLOT 0,88: DRAW 255,0
```

Si tracciano quindi i nuovi assi che rispetto al sistema di riferimento dell'elaboratore avranno origine:

asse x dal punto (0,88): linea 35

asse y dal punto (128,0): linea 30

Le prime istruzioni delle linee 30 e 35 (PLOT) definiscono il punto da cui origina il segmento, le seconde (DRAW) lo tracciano.

Proseguendo nel programma, con un'iterazione viene graduato l'asse x per mezzo di piccoli segmenti verticali a distanza di 10 unità l'uno dall'altro.

Per ogni valore di i (linea 40), l'ascissa del punto d'origine del segmento incrementerà di 10 e l'ordinata rimarrà costantemente fissa ad 87, un'unità in meno di quella a cui si trova l'asse delle ascisse ($y = 88$).

Con la seconda istruzione della linea 50 si traccia un segmento verticale che, partendo dal punto assegnato, ne incrementerà l'ordinata di 2 unità.

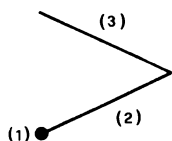
```
40>FOR i=1 TO 24
50 PLOT i*10-2,87: DRAW 0,2
60 NEXT i
```

Nell'istruzione PLOT $i*10-2,87$, la correzione -2 si rende necessaria per far coincidere la graduazione dell'asse di 10 in 10 con i valori assegnati al nuovo sistema di riferimento.

Ad esempio: l'origine ora si trova a (128,88), la graduazione degli assi deve passare per questo punto: con la variabile $i=13$ si otterrebbe, senza la correzione, $i*10=130$, ma l'origine ha ascissa 128, due unità in meno rispetto al risultato del calcolo.

Per disegnare il verso (la freccia) dell'asse x, le istruzioni sono:

```
65>PLOT 251,85: DRAW 4,3: DRAW
-4,3
```



Il primo ordine sulla linea 65 (1) posiziona il punto di partenza
il secondo (2) traccia il primo segmento,
il terzo (3) completa la freccia.

Con una nuova iterazione viene graduato l'asse y (linee 70-80-90) e ne viene disegnato il verso (linea 95) con un procedimento analogo a quello usato per l'asse x.

```
70>FOR i=1 TO 16
80 PLOT 127,i*10-2: DRAW 2,0
90 NEXT i
95 PLOT 125,170: DRAW 3,4: DRA
W 3,-4
```

È necessario ora sistemare sul video la posizione dell'origine 0 e visualizzare l'unità di misura:

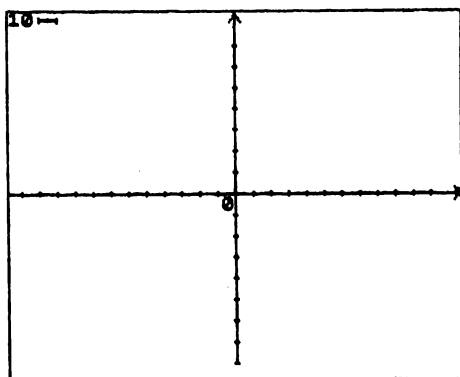
```
97>PRINT AT 11,15;0: PRINT AT
0,0;10: PLOT 18,171: DRAW 10,0:
DRAW 0,-1: DRAW 0,2: PLOT 18,170
: DRAW 0,2
```

- 1) PRINT AT 11,15;0 usando il sistema di riferimento dell'elaboratore per il modo di scrittura NORMALE (pagina PRINT), si scrive nell'undicesima riga e quindicesima colonna lo 0 in corrispondenza dell'origine.
- 2) PRINT AT 0,0;10 sullo schermo in alto a sinistra viene scritto il numero di unità contenute nell'unità di misura grafica che viene usata per graduare gli assi.
- 3) PLOT 18,171: DRAW 10,0 a fianco del numero 10 viene tracciato un segmento lungo 10 u
- 4) DRAW 0,1: DRAW 0,2 all'estremo destro del segmento si disegna un tratto breve verticale (2u) per delimitarlo
- 5) PLOT 18,170: DRAW 0,2 si riporta all'estremo sinistro e si disegna un uguale tratto verticale di delimitazione.

La linea di programma che esegue tutte queste istruzioni risulta quindi la seguente

```
97>PRINT AT 11,15;0: PRINT AT
0,0;10: PLOT 18,171: DRAW 10,0:
DRAW 0,-1: DRAW 0,2: PLOT 18,170
: DRAW 0,2
```

Facendo girare il programma con RUN, si ottiene:



Esercizi

1. Prova a costruire un sistema di riferimento cartesiano di 4 quadranti in cui l'origine è posta nel punto di coordinate (50,70) o in altri punti a piacimento.

```
30>PLOT 50,0: DRAW 0,175
35 PLOT 0,70: DRAW 255,0
```

Sostituire nel programma ogni volta che compare 128 la nuova ascissa dell'origine e ogni volta che compare 88 la nuova ordinata dell'origine.

2. Prova a graduare gli assi di 20 in 20 nel sistema di assi cartesiani in cui l'origine è in (128,88).

```
40>FOR i=1 TO 13
50 PLOT i*20-12,87: DRAW 0,2
60 NEXT i
70 FOR i=1 TO 9
80 PLOT 127,i*20-12: DRAW 2,0
90 NEXT i
```

_____ FATTORE DI CORREZIONE

3. Prova a graduare gli assi di 50 in 50.

```
40>FOR i=1 TO 5
50 PLOT i*50-22,87: DRAW 0,2
60 NEXT i
70 FOR i=1 TO 3
80 PLOT 127,i*50-12: DRAW 2,0
90 NEXT i
```

_____ FATTORE DI CORREZIONE

2.3.2 Inserimento punti

È importante tener presente che rispetto al sistema di riferimento dell'elaboratore, è stata effettuata una traslazione

$$x_1 = x_0 + 128 \text{ e } y_1 = y_0 + 88$$

dove (x_0, y_0) sono le coordinate rispetto al sistema dell'elaboratore e (x_1, y_1) sono quelle riferite al piano cartesiano disegnato sul video.

Quindi ogni punto inserito sulla base del nuovo piano cartesiano (di 4 quadranti) che si ha sul video, dovrà essere incrementato di 128 per l'ascissa e di 88 per l'ordinata.

```
100>REM **inserimento punti**
110 INPUT "inserisci le coordin
ate del punto :";x;"",y
120>PLOT x+128,y+88
```

Per mezzo di un INPUT (linea 110) si assegna all'indirizzo di memoria x l'ascissa del punto che si desidera inserire e alla memoria y, la corrispondente ordinata. Con la linea 120 è automatica la traslazione di 128 in ascissa ed 88 in ordinata, che tramuta le coordinate del punto inserito rispetto al piano cartesiano presente sul video in quelle del sistema di riferimento utilizzato dall'elaboratore.

In questo modo il calcolatore accetta anche valori negativi per le coordinate che nel nuovo sistema di riferimento possono variare tra

$$-127 \leq x \leq 127 \text{ e } -87 \leq y \leq 87$$

Esiste la possibilità che l'operatore inserisca per errore valori esterni al campo di variabilità di x e y, in questo caso il programma si ferma alla 1ª istruzione della linea 120 segnalando errore B

B Integer out of range, 120:1

Si può ovviare a questo inconveniente aggiungendo una linea di programma precedente l'istruzione di PLOT.

```
115>IF x<=-127 OR x>=127 OR y<=-87 OR y>=87 THEN GO TO 110
```

Basta cioè che si verifichi una sola delle circostanze considerate perché si ritorni alla linea di INPUT.

La linea successiva del programma stampa a fianco del punto disegnato sullo schermo la lettera A.

```
130>PRINT AT 22-INT (y+88)/8,INT (x+128)/8;"A"
```

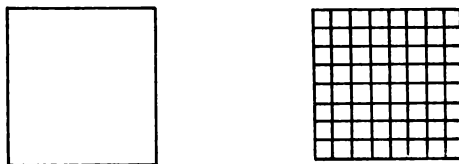
Dal momento che i caratteri dell'alfabeto vengono posizionati sul video secondo il reticolo della pagina PRINT è necessario usare un'espressione aritmetica di conversione:

per determinare la colonna: $\text{INT} (x+128)/8$

per determinare la riga: $22-\text{INT} (y+88)/8$

— INT è la funzione dell'elaboratore che permette di ottenere la parte intera di un numero (arrotondato sempre per difetto)

— entrambe le espressioni sono divise per 8 perché il pixel della pagina PRINT equivale a 8x8 punti della pagina PLOT.



— $22-\text{INT} (y+88)/8$: la complessità di quest'espressione è dovuta al fatto che il pixel 0,0 in pagina PRINT si trova in alto a sinistra mentre il punto di coordinate (0,0) della pagina PLOT si trova in basso a sinistra.

Per poter situare la lettera A di fianco al punto (x,y) alla riga corretta è necessario:

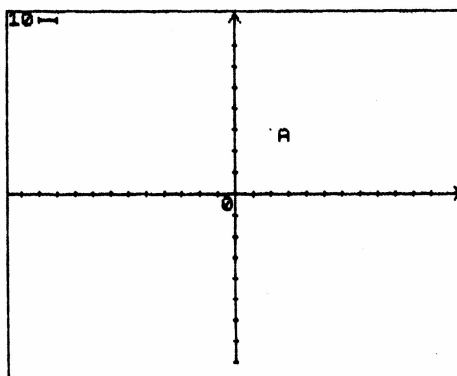
a) scendere per far collimare il pixel di partenza con il punto (0,0), quest'ultimo si troverà nella ventiduesima riga della pagina PRINT:



b) risalire con una traslazione verticalmente fino alla riga $\text{INT} (y+88)/8$ su cui si trova il punto di ordinata (y+88)

— $(x+128)$ è la consueta espressione (vedi punto b) analizzato in precedenza che permette di traslare la lettera lungo l'asse x nel nuovo sistema di riferimento

Facendo girare il programma con RUN ottengo:



Esercizi:

4. Come dovrà essere modificata la linea 120 per consentire gli stessi automatismi di traslazione quando l'unità di misura del nuovo sistema di riferimento è doppia rispetto a prima?
 R. $\text{PLOT INT } (x/2+128), \text{INT } (y/2+88)$
 Il sistema perde in precisione? Perché?
 R. sì, poiché inserendo coordinate dispari si otterranno risultati decimali che verranno sistematicamente arrotondati per difetto a causa della funzione INT.
5. Prova a costruire sistemi di riferimento con unità grafiche di misura diverse e modifica di conseguenza la linea 120 e la linea 97 in cui viene visualizzata l'U.M.
 R. vedi es 2-3, con U.M. 50:

```
97 PRINT AT 11,15;0:PRINT AT 0,0
;50
120 PLOT INT (x/5+128), INT (y/5
+88)
```

Potrebbe rivelarsi utile inserire più punti denominati con lettere successive dell'alfabeto. Per fare ciò è sufficiente aggiungere le seguenti linee al programma

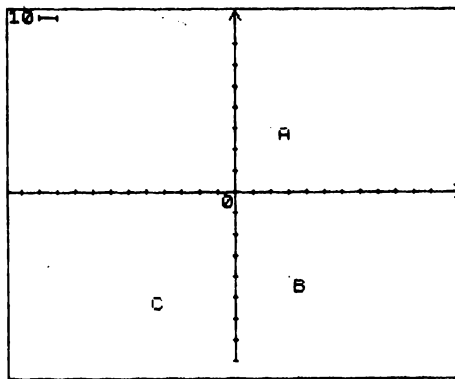
```
105 INPUT "Quanti punti vuoi in
serie ? (max 21)";f
107 FOR i=1 TO f
125 READ a$
135 NEXT i
137 RESTORE
1000 DATA "A","B","C","D"
```

e modificare le linee 115 e 130 in questo modo

```
115 IF x<=-127 OR x>=127 OR y<=
-87 OR y>=87 THEN LET i=i-1: GO
TO 135
130 PRINT AT 22-INT (y+88)/8,IN
T (x+128)/8;a$
```

Con l'INPUT di linea 105 si definisce il numero di punti che l'operatore desidera disegnare: per mezzo di un'iterazione vengono inserite le coordinate e ad ognuna viene assegnata una lettera in successione alfabetica (linee 125-130).

Alla fine dell'iterazione il puntatore che veniva fatto scorrere lungo i DATA viene rimesso nella posizione di partenza, pronto per fornire ancora le lettere. Facendo girare nuovamente il programma ottengo:

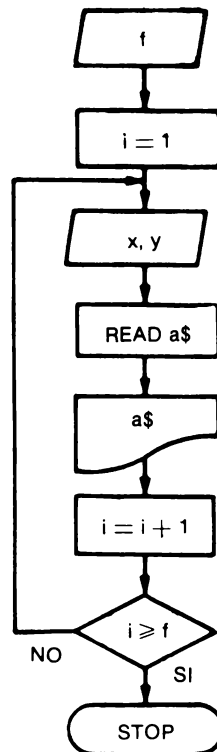


Il programma modificato risulta quindi:

```

100>REM **inserimento punti**
105 INPUT "Quanti punti vuoi in
serie ? (max21) "; f
107 FOR i=1 TO f
110 INPUT "Inserisci le coordin
ate del punto :";x(i);";";y(
i)
115 IF x(i)<=-127 OR x(i)>=127
OR y(i)<=-87 OR y(i)>=87 THEN LE
T i=i-1: GO TO 135
120 PLOT x(i)+128,y(i)+88
125 READ a$
130 PRINT AT 22-INT (y(i)+88)/8
,INT (x(i)+128)/8;a$
135 NEXT i
137 RESTORE
140 DATA "A","B","C","D"

```



2.4 Rappresentazione delle proiezioni dei punti tracciati sul piano cartesiano

Servizio U.D.S.

CODICE MAT 9/b

AUTORE PAOLA CATALANI

ANNO 1984

SCUOLA MEDIA STATALE "DI VITTORIO" - Pioltello (Mi)

TITOLO "Proiezione di punti su assi cartesiani"

RIASSUNTO

Si disegna la proiezione di uno o più punti sugli assi di un sistema cartesiano a quattro quadranti presente sul video

Curatori: PIERO D'ALFONSO - GIOVANNI MARTORANA

Elementare	<input type="checkbox"/>	Alfabetizzazione informatica	<input type="checkbox"/>
Media	<input checked="" type="checkbox"/>	linguistica	<input type="checkbox"/>
Media superiore	<input type="checkbox"/>	scienze / matematica	<input checked="" type="checkbox"/>
		tecnica	<input type="checkbox"/>
		grafica	<input type="checkbox"/>
BASIC	<input checked="" type="checkbox"/>	LOGO	<input type="checkbox"/>
		PASCAL	<input type="checkbox"/>
iterazione FOR/NEXT	<input checked="" type="checkbox"/>	(solo per BASIC)	
scelta condizionata IF/THEN	<input checked="" type="checkbox"/>		
matrici di memoria DIM	<input checked="" type="checkbox"/>		
READ/DATA	<input checked="" type="checkbox"/>		
DATA su FILE	<input type="checkbox"/>		
soubroutine GOSUB/RETURN	<input checked="" type="checkbox"/>		
menù	<input type="checkbox"/>		
memoria di schermo	<input type="checkbox"/>		
word processing	<input type="checkbox"/>		
collegamenti a rete	<input type="checkbox"/>		
.....			
SINCLAIR ZX 81	<input type="checkbox"/>		
SINCLAIR SPECTRUM	<input checked="" type="checkbox"/>	MINIDRIVE	<input type="checkbox"/>
SINCLAIR QL	<input type="checkbox"/>		
COMMODORE VIC 20	<input type="checkbox"/>	DRIVE PER FLOPPY	<input type="checkbox"/>
COMMODORE C 64	<input type="checkbox"/>	DRIVE PER FLOPPY	<input type="checkbox"/>
M 10 OLIVETTI	<input type="checkbox"/>		
M 20 OLIVETTI	<input type="checkbox"/>	DRIVE	<input type="checkbox"/>
SHARP MZ 700	<input type="checkbox"/>	DRIVE	<input type="checkbox"/>
APPLE II	<input type="checkbox"/>	DRIVE PER FLOPPY	<input type="checkbox"/>
I.B.M. PC	<input type="checkbox"/>	DRIVE	<input type="checkbox"/>

Progetto "Banca delle Unità Didattiche"

Nell'utilizzare il piano cartesiano capita spesso di aver bisogno di tracciare la proiezione di punti sull'asse x e y.

Conviene allora utilizzare il programma relativo alla costruzione del piano cartesiano e completarlo programmandolo per tracciare la proiezione dei punti sugli assi.

È conveniente far chiedere all'elaboratore se si desiderano queste proiezioni (linea 160).

```

160>INPUT "vuoi le proiezioni d
ei punti sugli assi? (s/n)";
170 IF $="s" THEN GO SUB 250
180 STOP

```

Se la risposta è positiva (s) si rimanda ad un sottoprogramma che inizia alla linea 250, altrimenti il programma si ferma con l'istruzione: 180 STOP.

Per ogni punto si deve tracciare:

- 1) la proiezione sull'asse x;
- 2) la proiezione sull'asse y.

Per ogni proiezione si devono considerare due possibilità:

- 1) coordinata > 0;
- 2) coordinata < 0.

```

250>REM *proiezione sugli assi*
260 IF x>0 THEN GO TO 268
263 FOR i=x TO 0 STEP 3: GO TO
270
268 FOR i=0 TO x STEP 3
270 PLOT i+120,y+88
280 NEXT i

```

Con questa serie di istruzioni si traccia la proiezione sull'asse delle ascisse del punto A (x,y).

Se il valore di x è positivo il punto si trova sopra l'asse, la proiezione partirà dal punto A e scenderà fino all'asse x (linea 263), in caso contrario dovrà risalire verso l'alto (linea 268): da qui la necessità della scelta condizionata posta in linea 260.

Il segmento di proiezione viene tracciato usando un'iterazione (variabile di controllo i) per sistemare una serie di punti e non, come sarebbe più ovvio, utilizzando la prola chiave DRAW.

Questa complicazione è necessaria per disegnare una linea tratteggiata che viene realizzata utilizzando nell'iterazione STEP3 producendo cioè un incremento di 3 unità per volta nell'iterazione: si disegna un punto e se ne saltano tre.

Stesso ragionamento per la proiezione sull'asse y.

```

290>IF y>0 THEN GO TO 298
293 FOR i=y TO 0 STEP 3: GO TO
300
298 FOR i=0 TO y STEP 3
300 PLOT x+120,i+88
310 NEXT i
380>RETURN

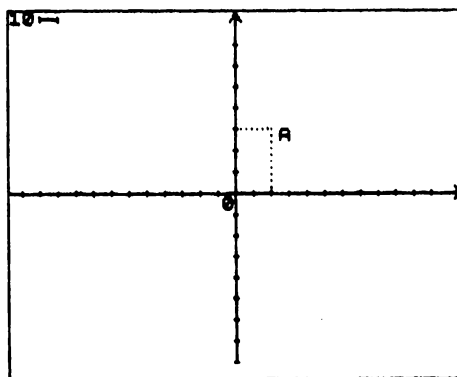
```

Utilizzando un'analoga procedura:

le linee 290 - 310 tracciano la proiezione sull'asse y del punto A

la linea 380 serve per tornare sul programma principale.

Facendo nuovamente girare il programma con RUN, a questo punto si ottiene:



Esercizi: Prova a modificare le linee 268 e 298 per cambiare il tipo di tratteggio
R. modificare il valore di STEP

Se si desidera avere la proiezione di più punti presenti contemporaneamente sul video è necessario modificare il programma.

Partendo dalle linee 100 - 130 del programma base modificato (UDS MAT 8/a)

```
100>REM **inserimento punti**
105 INPUT "Quanti punti vuoi in
serie ? (max21) ";f
107 FOR i=1 TO f
110 INPUT "Inserisci le coordin
ate del punto :";x(i);";";y(i)
115 IF x(i)<=-127 OR x(i)>=127
OR y(i)<=-87 OR y(i)>=87 THEN LE
T i=i-1: GO TO 135
120 PLOT x(i)+128,y(i)+88
125 READ a$
130 PRINT AT 22-INT (y(i)+88)/8
,INT (x(i)+128)/8;a$
135 NEXT i
```

è necessario apportare una serie di modifiche e aggiunte:

— innanzi tutto serve aprire una dimensione di memoria che immagazzinerà i valori dell'ascissa (DIM x) e dell'ordinata (DIM y) di ognuno degli f punti inseriti:

106 DIM x (f): DIM y (f)

— successivamente durante l'iterazione per l'inserimento dei punti (linee 110 - 135), si devono modificare le linee:

```
110 INPUT "inserisci le coordinate del punto: "; x(i);";";y(i)
120 PLOT x(i) + 128, y(i) + 88
```

in modo da indirizzare correttamente in memoria i valori assegnati alle variabili x e y.

Le istruzioni necessarie per tracciare le proiezioni sugli assi dovranno essere ripetute per ogni punto, quindi si rende necessaria un'iterazione che faccia ripetere f volte l'operazione.

```
255 FOR n = 1 TO f
315 NEXT n
```

Infine tutte le linee comprese tra 260 e 300 devono essere modificate mettendo in evidenza l'indice per i valori di x e y in conseguenza del dimensionamento della memoria, quindi tutte le x diventeranno x (n) e le y, y(n).

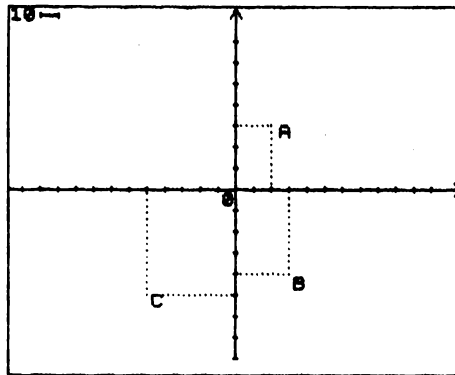
Ricapitolando, il programma è il seguente:

```
100>REM **inserimento punti**
105 INPUT "Quanti punti vuoi in
serie ? (max21) ";f
106 DIM x (f): DIM y (f)
107 FOR i=1 TO f
110 INPUT "Inserisci le coordin
ate del punto :";x(i);";";y(i)
115 IF x(i)<=-127 OR x(i)>=127
OR y(i)<=-87 OR y(i)>=87 THEN LE
T i=i-1: GO TO 135
120 PLOT x(i)+128,y(i)+88
125 READ a$
130 PRINT AT 22-INT (y(i)+88)/8
,INT (x(i)+128)/8;a$
135 NEXT i
137 RESTORE
140 DATA "A","B","C","D"
150 INPUT "Vuoi le proiezioni? (
s/n) ";a$
170 IF a$="s" THEN GO SUB 250
```

```

250>REM **proiezioni sugli assi
**
255 FOR n=1 TO f
260 IF x(n)>0 THEN GO TO 265
263 FOR i=x(n) TO 0 STEP 3: GO
TO 270
265 FOR i=0 TO x(n) STEP 3
270 PLOT i+128,y(n)+88
280 NEXT i
290 IF y(n)>0 THEN GO TO 295
293 FOR i=y(n) TO 0 STEP 3: GO
TO 300
295 FOR i=0 TO y(n) STEP 3
300 PLOT x(n)+128,i+88
310 NEXT i
315 NEXT n
320 RETURN

```



Esercizi:

— Quali linee vanno aggiunte al programma per disegnare il segmento AB?

R. vanno aggiunte le linee:

```

500>LET a=x(2)-x(1): LET b=y(2)
-y(1)
510 PLOT x(1)+128,y(1)+88
520 DRAW a,b

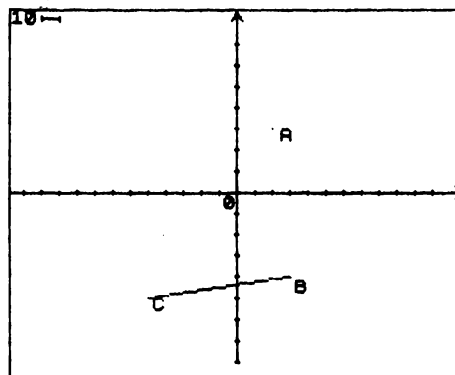
```

— e per disegnare il segmento BC?

```

500>LET a=x(3)-x(2): LET b=y(3)
-y(2)
510 PLOT x(2)+128,y(2)+88
520 DRAW a,b

```



2.5 Traslazione di un punto sul piano equazione di traslazione

Servizio U.D.S.

CODICE	MAT 10/C
AUTORE	PAOLA CATALANI
ANNO	1984
SCUOLA	MEDIA STATALE "DI VITTORIO" - Pioltello (Mi)
TITOLO	"Traslazione di un punto sul piano"

RIASSUNTO

Vengono inserite le coordinate di A e A' e vengono richieste all'operatore le equazioni di traslazione, viene inoltre disegnato il vettore di traslazione

Curatori: PIERO D'ALFONSO - GIOVANNI MARTORANA

Elementare	<input type="checkbox"/>	Alfabetizzazione informatica	<input type="checkbox"/>
Media	<input checked="" type="checkbox"/>	linguistica	<input type="checkbox"/>
Media superiore	<input type="checkbox"/>	scienze / matematica	<input checked="" type="checkbox"/>
		tecnica	<input type="checkbox"/>
		grafica	<input type="checkbox"/>
BASIC	<input checked="" type="checkbox"/>	LOGO	<input type="checkbox"/>
		PASCAL	<input type="checkbox"/>
iterazione FOR/NEXT	<input checked="" type="checkbox"/>	(solo per BASIC)	
scelta condizionata IF/THEN	<input checked="" type="checkbox"/>		
matrici di memoria DIM	<input type="checkbox"/>		
READ/DATA	<input type="checkbox"/>		
DATA su FILE	<input type="checkbox"/>		
soubroutine GOSUB/RETURN	<input checked="" type="checkbox"/>		
menù	<input type="checkbox"/>		
memoria di schermo	<input type="checkbox"/>		
word processing	<input type="checkbox"/>		
collegamenti a rete	<input type="checkbox"/>		
.....			
SINCLAIR ZX 81	<input type="checkbox"/>		
SINCLAIR SPECTRUM	<input checked="" type="checkbox"/>	MINIDRIVE	<input type="checkbox"/>
SINCLAIR QL	<input type="checkbox"/>		
COMMODORE VIC 20	<input type="checkbox"/>	DRIVE PER FLOPPY	<input type="checkbox"/>
COMMODORE C 64	<input type="checkbox"/>	DRIVE PER FLOPPY	<input type="checkbox"/>
M 10 OLIVETTI	<input type="checkbox"/>		
M 20 OLIVETTI	<input type="checkbox"/>	DRIVE	<input type="checkbox"/>
SHARP MZ 700	<input type="checkbox"/>	DRIVE	<input type="checkbox"/>
APPLE II	<input type="checkbox"/>	DRIVE PER FLOPPY	<input type="checkbox"/>
I.B.M. PC	<input type="checkbox"/>	DRIVE	<input type="checkbox"/>

Progetto "Banca delle Unità Didattiche"

Lo scopo di questa UDS è quello di individuare il vettore di traslazione per mezzo delle equazioni che mettono in corrispondenza i punti A, con il punto traslato A', le coordinate dei quali sono state inserite entrambe dall'operatore.

Per ottenere il disegno di un punto sul piano cartesiano è necessario:

- 1) tracciare il sistema di riferimento (4 quadranti): vedi UDS MAT 8/a (linee 10 - 97);
- 2) inserire le coordinate del punto, riferite al sistema presente sullo schermo: vedi UDS MAT 8/a (linee 100-130);
- 3) inserire le coordinate del punto traslato A' con una serie di istruzioni analoghe a quelle utilizzate per sistemare sul video il punto A, in cui le coordinate di A' sono rispettivamente x' e y'.

```

140 INPUT "Inserisci le coordin
ate del punto traslato";x
    y1
145 PLOT x1+128,y1+88
150 PRINT AT 22-INT (y1+88)/8,I
NT (x1+128)/8,"A'"
155 IF x=x1 AND y=y1 THEN PRINT
AT INT (y-88)/8,INT (x+128)/8;"
A=A'"

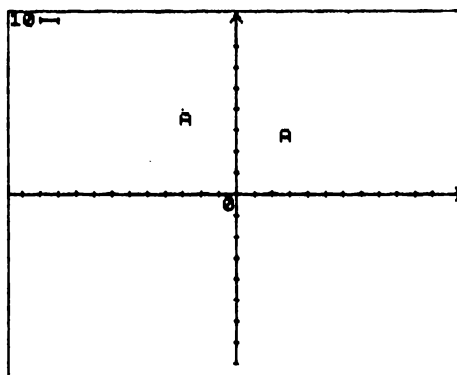
```

La linea 155 considera l'eventualità che i due punti siano coincidenti.

Senza queste istruzioni l'elaboratore sovrapporrebbe la scrittura A' ad A ed il punto A non sarebbe più chiaramente definito sul piano.

Con quest'aggiunta, in caso di coincidenza, accanto al punto, compare la scritta A=A'.

Facendo girare il programma con RUN si ottiene:



Per individuare le equazioni di traslazione è molto utile tracciare le proiezioni dei punti A e A' sugli assi (vedi UDS MAT 9/b: linee 160-170 e 250-310).

Con una serie di istruzioni analoghe a quelle utilizzate precedentemente vengono tracciate le proiezioni per il punto traslato A'.

```

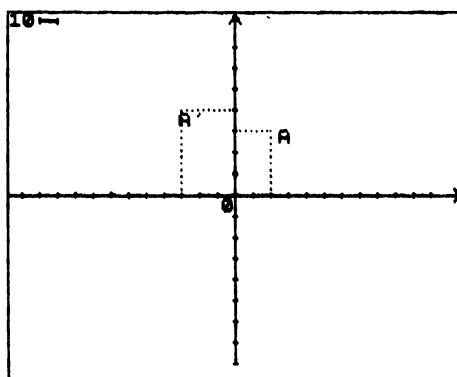
320 IF x1>0 THEN GO TO 328
323 FOR i=x1 TO 0 STEP -3: GO TO
330
328 FOR i=0 TO x1 STEP 3
330 PLOT i+128,y1+88
340 NEXT i
350 IF y1>0 THEN GO TO 358
353 FOR i=0 TO y1 STEP -3: GO T
O 360
358 FOR i=0 TO y1 STEP 3
360 PLOT x1+128,i+88
370 NEXT i
380 RETURN

```

Le linee 320-340 tracciano la proiezione sull'asse x di A'.

Le linee 340-370 tracciano la proiezione sull'asse y di A'.

Facendo girare il programma con RUN si ottiene:



A questo punto ritornando sul programma principale, vengono inserite le equazioni di traslazione:

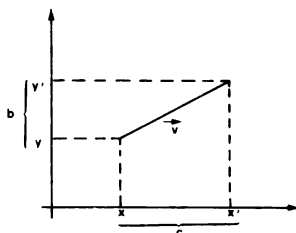
```
200>PRINT AT 0,0;"A(";"x";";";y";"  
210 FOR i=1 TO 3: INPUT "Inseri  
   sci le equazioni di traslazione  
x'=x+c; y'=y+b  
220 GO TO 500  
230 NEXT i
```

Con le istruzioni della linea 200, l'elaboratore scrive in alto a sinistra (AT 0,0) le coordinate dei punti, necessarie all'operatore per poter inserire le equazioni mediante l'INPUT di linea 210, con un'iterazione si fissa a tre il numero massimo di tentativi a disposizione dell'utente per l'inserimento.

La linea 220 rimanda al sottoprogramma in cui si confronteranno i valori inseriti con quelli reali: terminato il confronto il programma prosegue continuando l'iterazione (linea 230).

Negli indirizzi di memoria c e b vengono poste le componenti del vettore di traslazione rispettivamente lungo l'asse x e y

Equazione di traslazione
di vettore V
 $X' = X + C$
 $Y' = Y + B$



c e b sono le intensità delle componenti del vettore di traslazione lungo gli assi ed hanno segno positivo o negativo a seconda che siano concordi con gli assi oppure no.

Se il movimento del punto è avvenuto parallelamente ad un asse il valore di c o b è 0: in questo caso è necessario esplicitare questo valore, altrimenti l'elaboratore non si sposterà dalla fase di INPUT.

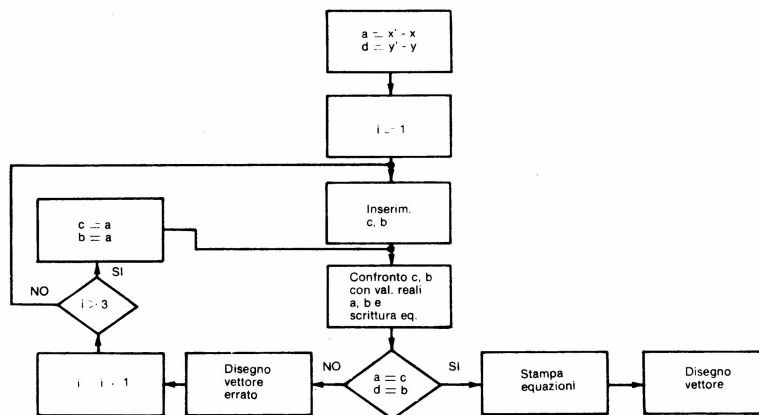
La linea 220 rimanda ad un sottoprogramma che inizia alla linea 490

```
490>REM **equazioni di traslazi  
one**  
500 LET a=x1-x  
510 LET d=y1-y
```

e assegna agli indirizzi di memoria a (nella linea 500) e d (nella linea 510) i valori reali delle componenti del vettore di traslazione, rispettivamente, lungo l'asse x e y.

A questo punto il programma deve confrontare i valori immessi dall'operatore (c,b) con quelli reali memorizzati (a,d) e, se sono esatti, stampare il sistema di equazione per esteso.

Con il seguente diagramma di flusso a blocchi viene schematizzata la sequenza di operazioni svolta dall'elaboratore.



Esiste però un problema di scrittura nelle equazioni

— se a o d sono negativi con l'istruzione:

(1) PRINT "x' = x+"; a

con a = -5 ad esempio ottengono sul video: x'=x+-5, espressione matematicamente scorretta.

Modificando l'istruzione in:

(2) PRINT "x'=x"; a

con a=-5 ottengo "x'=x-5

— Però, con a o d positivi, l'uso dell'istruzione (2) porta ad ottenere:

es: a=3 dà x'=x3 espressione ancora scorretta.

È evidente quindi la necessità di programmare due strategie di scrittura:

1°) con a o d negativi: PRINT "x'=x";a

2°) con a o d positivi: PRINT "x'=x+";a

È necessario analizzare caso per caso le combinazioni possibili:

```

530>IF a=c AND a>=0 THEN PRINT
AT 19,0,"x'=x+",a: GO TO 550
  
```

Se il valore c inserito dall'operatore per la componente del vettore lungo l'asse x è uguale al valore reale a, e se a è positivo (caso 2°) l'elaboratore stampa l'equazione nella forma matematica corretta e va direttamente, con una scelta incondizionata (GOTO) a confrontare il secondo valore (componente del vettore lungo l'asse y) alla linea 550:

```

540>IF a=c AND a<0 THEN PRINT A
T 19,0,"x'=x";a: GO TO 550
  
```

l'elaboratore effettua le stesse operazioni della linea precedente, ma stampa l'espressione corretta nel caso in cui il valore di a sia negativo (caso 1°):

```

545>GO TO 610
  
```

rimanda alla linea 600 nel caso in cui il valore di c inserito dall'operatore risulti errato:

```

550>IF d=b AND d>=0 THEN PRINT
AT 20,0,"y'=y+",b: GO TO 580
  
```

esegue il confronto come nella linea 530 tra i valori di a e b, cioè l'ampiezza della traslazione lungo l'asse y, con d positivo (caso 2°):

```
550>IF d=b AND d<0 THEN PRINT A
T 20,0;"y'=y",b: GO TO 580
```

come nella linea 540: ampiezza lungo l'asse y: valore di a negativo (caso 1°):

```
570>GO TO 610
```

rimanda alla linea 600 nel caso in cui l'operatore abbia inserito la prima equazione corretta e la seconda errata:

```
610>IF i=3 THEN LET c=a: LET b=
d: PRINT AT 19,0;"le equazioni e
rano: " : GO TO 530
615 GO SUB 700
620>RETURN
```

Non trovando la linea 600 l'elaboratore prosegue a quella successiva.

La linea 610 contiene l'istruzione di corretta assegnazione dopo i tre tentativi.

Le istruzioni che seguono sulla stessa linea sono condizionate da IF... THEN all'assegnazione automatica dei valori corretti.

IF i=3 THEN LET c=a: LET b=d se dopo tre tentativi l'utente non ha ancora inserito i valori di c e b corretti, a questo punto automaticamente con le due istruzioni di LET verranno sostituiti negli indirizzi di memoria c e b i valori esatti, cioè a e d.

PRINT AT 19,0;"...." sempre dopo tre tentativi si stamperà sul video "le equazioni erano"

GOTO 530 fa rifare automaticamente il confronto, che questa volta risulterà ovviamente esatto. Verranno cioè stampate le due equazioni corrette.

Se invece l'operatore ha inserito valori errati, ma non ancora per tre volte, l'elaboratore passa alla linea successiva 615 che lo manda alla subroutine "COSTRUZIONE VETTORE".

Questa parte del programma verrà analizzata successivamente, per il momento basti dire che sul video sarà tracciato il vettore errato, utilizzando cioè i dati non corretti immessi dall'operatore che così potrà rendersi conto dell'errore.

Alla fine della subroutine ci sarà la parola chiave RETURN che automaticamente farà tornare l'elaboratore alla linea successiva il GOSUB relativo, cioè a 630.

Con la linea 620 si conclude il sottoprogramma che consente all'utente tre possibilità di inserimento, il programma cioè ricomincia dalla fase di INPUT delle equazioni alla linea 210.

Si deve ancora considerare il caso in cui il confronto risulti positivo, questo accade:

- quando l'operatore inserisce valori di c e b corretti
- dopo il terzo tentativo quando l'elaboratore inserisce automaticamente in memoria i valori esatti:

```
580>GO SUB 700
590 STOP
```

Anche in questo caso il programma prosegue con la costruzione del vettore per mezzo della linea 580.

Disegnato il vettore, questa volta corretto, il programma, rimandato alla linea successiva GOSUB, si ferma alla linea 590.

Ricapitolando, questa parte del programma si articola nelle linee del listato seguente.

```
490>REM **equazioni di traslazi
one**
500 LET a=x1-x
510 LET d=y1-y
530 IF a=c AND a>=0 THEN PRINT
AT 19,0;"x'=x+",a: GO TO 550
540 IF a=c AND a<0 THEN PRINT A
T 19,0;"x'=x",a: GO TO 550
545 GO TO 610
550 IF d=b AND d>=0 THEN PRINT
AT 20,0;"y'=y+",b: GO TO 580
560 IF d=b AND d<0 THEN PRINT A
T 20,0;"y'=y",b: GO TO 580
570 GO TO 610
580 GO SUB 700
590 STOP
610 IF i=3 THEN LET c=a: LET b=
d: PRINT AT 19,0;"le equazioni e
rano: " : GO TO 530
615 GO SUB 700
620 RETURN
```

Esercizi

1) Come va modificato il programma per dare all'utente un diverso numero di possibilità d'errore, ad esempio 5

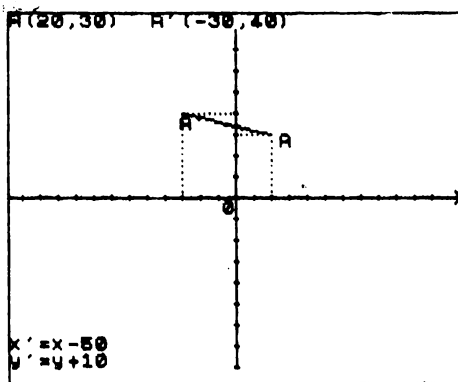
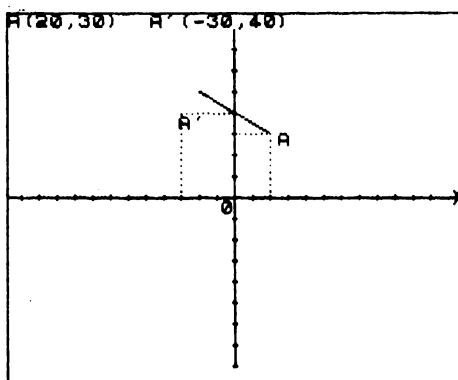
R. 210 FOR i = 1 TO 5: INPUT...

2.5.1 Costruzione vettore

```
700>REM **vettore di traslazio  
ne**  
720 PLOT x+128,y+88: DRAW c,b  
750 RETURN
```

È sufficiente una sola linea di programma in cui viene tracciato un segmento che partendo dal punto A (x,y) arriverà al punto di coordinate $(x \pm c, y \pm b)$ che corrispondono esattamente a quelle di A' nel caso in cui i valori inseriti dall'operatore siano corretti; in caso contrario verrà tracciato un vettore errato e l'utente si accorgerà dell'errore dal disegno sullo schermo.

Facendo girare il programma si ottiene:



Esercizio

Come far apparire di colore rosso il vettore di traslazione quando è errato ed in nero nel caso in cui sia esatto?

R. Aggiungendo le 2 linee:

613 INK 2

730 INK 0

2.6 Disegno di figure sul piano cartesiano

Servizio U.D.S.

CODICE 11/d

AUTORE PAOLA CATALANI

ANNO 1984

SCUOLA MEDIA STATALE "DI VITTORIO" - Pioltello

TITOLO "Disegno figure su piano cartesiano"

RIASSUNTO

Viene costruito un piano cartesiano (4 quadranti) su una parte del video e viene disegnata una figura qualsiasi in base alle coordinate dei vertici inserite dall'operatore.

Curatori: PIERO D'ALFONSO - GIOVANNI MARTORANA

Elementare	<input type="checkbox"/>	Alfabetizzazione informatica	<input type="checkbox"/>
Media	<input checked="" type="checkbox"/>	linguistica	<input type="checkbox"/>
Media superiore	<input type="checkbox"/>	scienze / matematica	<input checked="" type="checkbox"/>
		tecnica	<input type="checkbox"/>
		grafica	<input type="checkbox"/>
BASIC	<input checked="" type="checkbox"/>	LOGO	<input type="checkbox"/>
		PASCAL	<input type="checkbox"/>
iterazione FOR/NEXT	<input checked="" type="checkbox"/>	(solo per BASIC)	
scelta condizionata IF/THEN	<input type="checkbox"/>		
matrici di memoria DIM	<input checked="" type="checkbox"/>		
READ/DATA	<input checked="" type="checkbox"/>		
DATA su FILE	<input type="checkbox"/>		
soubroutine GOSUB/RETURN	<input checked="" type="checkbox"/>		
menù	<input type="checkbox"/>		
memoria di schermo	<input type="checkbox"/>		
word processing	<input type="checkbox"/>		
collegamenti a rete	<input type="checkbox"/>		
.....			
SINCLAIR ZX 81	<input type="checkbox"/>		
SINCLAIR SPECTRUM	<input checked="" type="checkbox"/>	MINIDRIVE	<input checked="" type="checkbox"/>
SINCLAIR QL	<input type="checkbox"/>		
COMMODORE VIC 20	<input type="checkbox"/>	DRIVE PER FLOPPY	<input type="checkbox"/>
COMMODORE C 64	<input type="checkbox"/>	DRIVE PER FLOPPY	<input type="checkbox"/>
M 10 OLIVETTI	<input type="checkbox"/>		
M 20 OLIVETTI	<input type="checkbox"/>	DRIVE	<input type="checkbox"/>
SHARP MZ 700	<input type="checkbox"/>	DRIVE	<input type="checkbox"/>
APPLE II	<input type="checkbox"/>	DRIVE PER FLOPPY	<input type="checkbox"/>
I.B.M. PC	<input type="checkbox"/>	DRIVE	<input type="checkbox"/>

Progetto "Banca delle Unità Didattiche"

Questa UDS si articola in tre parti

- 1) costruzione di un piano cartesiano (4 quadranti) sulla parte destra del video, lasciando a sinistra una finestra da utilizzare per la parte colloquiale;
- 2) inserimento delle coordinate dei vertici della figura;
- 3) disegno della figura sul piano cartesiano.

2.6.1 Costruzione del piano cartesiano

Per la comprensione approfondita del programma è auspicabile aver già lavorato con l'UDS MAT 8/a in quanto questa è un completamento dell'altra.

Nel caso in cui il programma preveda, in fase di INPUT, una parte colloquiale piuttosto lunga (maggiore di due righe) si rischia che la pagina video superiore subisca uno SCROLL (o scorrimento verso l'alto) automatico di una riga o più.

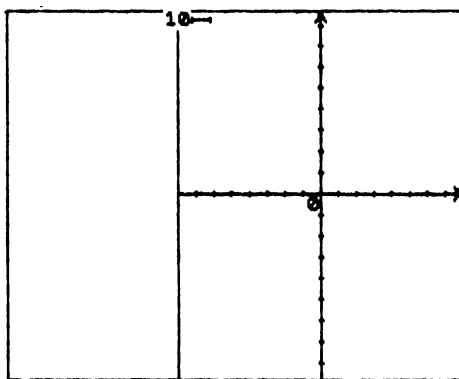
Questo è assolutamente da evitare per il fatto che sullo schermo compare un sistema di riferimento cartesiano fisso che, ovviamente, non può subire alcun spostamento (scroll).

Per evitare questo rischio esistono due possibilità:

- limitare la quantità di informazioni da dare all'utente in fase di INPUT rischiando però che l'operatore non capisca quale tipo di dati deve inserire.
- aprire una finestra su una parte del video in cui di volta in volta sia possibile scrivere tutte le informazioni necessarie, lasciando il sistema di riferimento sulla parte rimanente dello schermo.

In questo programma si è optato per la seconda soluzione.

La pagina video è stata divisa in due parti secondo il seguente schema:



La parte sinistra dello schermo viene utilizzata nelle prime 12 colonne della pagina PRINT per la parte colloquiale del programma; la parte destra serve invece, in configurazione PLOT, per costruire il sistema di riferimento cartesiano e disegnare le figure.

Le prime due linee del programma contengono le istruzioni per disegnare gli assi:

```
1000>REM **traslazione di figure
**
1010 REM **piano cartesiano**
1020 PLOT 176,0: DRAW 0,175
1030 PLOT 97,88: DRAW 158,0
```

La prima istruzione della linea 1020 fissa il primo punto dell'asse y, la seconda lo traccia.

In modo analogo si procede con la linea 1030 per l'asse x, solo che questa volta è necessario tener conto dell'esistenza della finestra larga 12 colonne: il punto di partenza dell'asse non sarà quindi ad ascissa 0 ma a 97, cioè $12 \cdot 8 + 1$.

Questa espressione aritmetica è la conseguenza del fatto che il pixel di pagina PRINT è formato da $8 \cdot 8$ punti di pagina PLOT: il numero delle colonne della finestra (12) va moltiplicato per i punti contenuti in ogni pixel delle righe orizzontali (8); l'aggiunta di un'unità serve semplicemente ad evitare che eventuali lettere presenti nella dodicesima colonna della finestra siano a contatto con il punto iniziale dell'asse. La seconda istruzione della linea 1030 (DRAW 158,0) traccia un segmento orizzontale lungo 158 unità.

Il valore di 158 è stato scelto in base al calcolo della differenza tra la larghezza massima di pagina PLOT (255) e l'ascissa del punto d'inizio (97): $255 - 97 = 158$.

Con una serie di istruzioni analoghe a quelle già utilizzate in UDS MAT 8/a, viene tracciata la graduazione degli assi di 10 in 10.

Eccone la descrizione sintetica.

Con un'iterazione, ripetuta per 15 volte, si disegna un piccolo segmento verticale di due unità (DRAW 0,2) sull'asse x

```
1040>FOR i=1 TO 15
1042 PLOT i*10+96,87: DRAW 0,2
1044 NEXT i
```

La prima istruzione della linea 1042 (PLOT i*10+96,87) definisce le coordinate del punto da cui incominciare a disegnare il segmento verticale.

L'espressione i*10 incrementa ogni volta di 10 unità il valore dell'ascissa cui però va aggiunta la costante 96 a causa della finestra di 12 colonne ($12 \cdot 8 = 96$): l'ordinata del punto da cui incominciare a tracciare il segmento di graduazione si troverà di un'unità al di sotto dell'asse x di equazione $y=88$ (secondo il sistema di riferimento dell'elaboratore).

La linea successiva del programma disegna il verso (freccia) dell'asse x

```
1047>PLOT 251,88: DRAW 4,3: DRAW
-4,3
```

Per graduare l'asse y si opera nello stesso modo:

```
1050>FOR i=1 TO 17
1053 PLOT 175,i*10-2: DRAW 2,0
1055 NEXT i
1057 PLOT 173,170: DRAW 3,4: DRA
W 3,-4
```

L'iterazione (linee 1050-1053-1055) disegna 17 segmenti orizzontali di due unità che partono tutti da punti di ascissa 175, un'unità in meno rispetto all'asse y, la rispettiva ordinata incrementerà di 10 in 10 ($i \cdot 10$): a questa espressione matematica va aggiunto il fattore di correzione -2 per far coincidere la graduazione con l'origine del sistema di riferimento già fissato sullo schermo.

La linea 1057 disegna il verso dell'asse y.

Con una serie di istruzioni viene scritto 0 all'origine e viene disegnata l'unità di misura usata per la graduazione degli assi.

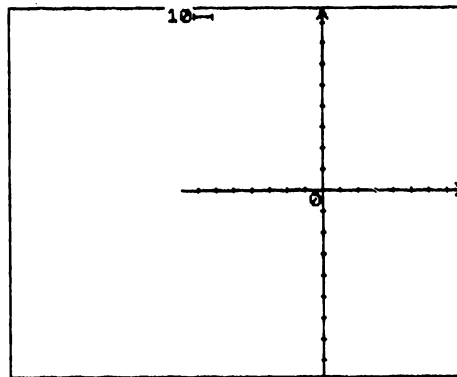
La linea 1060 scrive, secondo la pagina PRINT, all'11° riga e 21° colonna la cifra 0 in corrispondenza dell'origine degli assi.

```
1060>PRINT AT 11,21:0
1065 PRINT AT 0,11:10: PLOT 104,
171: DRAW 10,0: DRAW 0,-1: DRAW
0,2: PLOT 104,170: DRAW 0,2
```

La linea 1065 è costituita da numerose istruzioni che vanno analizzate singolarmente.

- PRINT AT 0,11;10 : sulla 1° riga, all'undicesima colonna viene scritto 10
- PLOT 104,171: DRAW 10,0 : traccia un segmento orizzontale di 10 unità (DRAW 10,0) partendo dal punto di coordinate (104,171), a fianco cioè del numero 10
- DRAW 0, —1; DRAW 0,2 : disegna un breve segmento verticale all'estremo destro del segmento indicante l'unità grafica
- PLOT 104,170: DRAW 0,2 : traccia un segmento verticale di delimitazione all'estremo sinistro dell'U.M.

Facendo girare il programma con RUN, si ottiene:



Esercizi

- 1 Prova a graduare gli assi di 20 in 20
 R. Vanno modificate le linee
 1040 FOR i= 1 TO 7
 1042 PLOT i*20+96,87: DRAW 0,2
 1050 FOR i=1 TO 9
 1053 PLOT 175, i*20-12: DRAW 2,0
 FATTORE DI CORREZIONE
- 2 Modifica il programma in modo da lasciare una finestra di 10 colonne, centrando il piano cartesiano nella parte rimanente del video
 R. Modificare le linee
 1020 PLOT 168,0: DRAW 0,175
 1030 PLOT 81,88: DRAW 170,0
 1040 FOR i= 1 TO 16
 1042 PLOT i*10+78,87: DRAW 0,2
 1053 PLOT 167, i*10-2: DRAW 2,0
 1057 PLOT 165,170: DRAW 3,4: DRAW 3, -4

2.6.2 Inserimento coordinate dei vertici della figura

Le coordinate dei vertici della figura che si desidera disegnare sul piano cartesiano vengono inserite con una serie di INPUT per mezzo di un'iterazione ed i valori di ciascuna vengono immessi in memoria x ed y: è necessario però che vengano mantenuti tutti i valori nella memoria dell'elaboratore e nell'ordine di inserimento, quindi serve dimensionare gli indirizzi di memoria x ed y (per mezzo della parola chiave DIM) sulla base del numero di coordinate che si desidera inserire, cioè dei vertici della figura.

Il programma quindi si svilupperà in questo modo:

```
1070>REM **inserimento punti**
1080 INPUT "Quanti vertici ha la
figura?";a
1085 DIM x(a): DIM y(a)
```

La linea 1080 rivolge all'operatore la domanda "quanti vertici ha la figura?": la risposta (variabile a) serve per dimensionare le memorie.

Nella linea successiva (1085) infatti vengono aperte due dimensioni di memoria, una per i valori delle ascisse DIM x, l'altra per quelli delle ordinate DIM y, di indice a: le due memorie cioè conterranno un numero di valori pari alla quantità dei vertici della figura, inoltre questi valori verranno immagazzinati esattamente nello stesso ordine con cui sono stati inseriti.

Ad esempio x(2), y(2) corrisponderanno alle coordinate x,y del secondo punto inserito.

Con una serie di INPUT l'operatore, a questo punto, dovrà inserire le coordinate dei vertici della figura: per poterla poi disegnare è necessario però che i punti vengano immessi seguendo il contorno della figura e non in ordine casuale.

Per poter comunicare queste informazioni all'utente viene utilizzata la finestra, cioè la parte a sinistra del video che era rimasta a disposizione.

Con una serie di istruzioni PRINT AT si scrive il messaggio su questa parte dello schermo avendo l'accortezza di far in modo che ogni riga non superi 12 lettere, cioè il numero di colonne che erano a disposizione:

```
1090>FOR n=1 TO a
1100>INPUT "Le coordinate: ";x(n)
      ";y(n)
1110>NEXT n
```

Per scrivere il messaggio vengono utilizzate le righe comprese tra la 13° e la 21°.

Con un'iterazione ripetuta per tante volte quanti sono i vertici della figura (a), vengono ora inserite le coordinate x e y dei vertici che saranno indirizzate rispettivamente nelle memorie x(n) ed y(n) dove la variabile n assume valori compresi tra 1 ed a, numero dei vertici.

```
1092>PRINT AT 13,0;"Dal vertice"
      PRINT AT 15,0;"in basso a"
1095>PRINT AT 17,0;"sin in senso"
      PRINT AT 19,0;"antiorario": P
      PRINT AT 21,0;"inserisci"
```

Dal momento che è possibile inserire diversi punti è comodo per l'operatore avere l'opportunità di leggere le coordinate dei punti già inseriti nella parte alta della finestra che era rimasta inutilizzata (il messaggio compare dalla 13° alla 21° riga).

Le seguenti linee del programma tutte inserite nell'iterazione servono a questo:

```
1102>READ a$
1104>PRINT AT n,0;a$;"(";x(n);",
      ";y(n);")"
1700>DATA "A","B","C","D","E","F"
      "G","H","I","L","M","N","O","P"
```

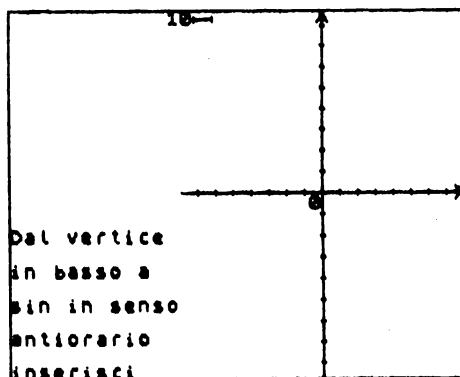
La linea 1102 fa leggere automaticamente all'elaboratore la \$ contenuta nei DATA di linea 1700: supponendo che il programma sia giunto a questa linea e stia percorrendo l'iterazione per la terza volta, il valore di a\$ sarà, in questo caso, C.

Probabilmente sarà opportuno ricordare che quando l'elaboratore incontra READ a\$ va automaticamente a cercare la stringa nel DATA che sta puntando, la immagazzina in memoria (nel primo caso "A") e sposta il puntatore sul successivo DATA (al di là della virgola). Quando arriva ad un nuovo READ a\$ immagazzinerà in memoria a\$ il nuovo data che sta puntando (in questo caso, "B").

La linea 1104 serve a far scrivere nella finestra, alla linea n partendo dal primo carattere a sinistra (0), la lettera dell'alfabeto che definisce il punto (a\$-seguita dalle coordinate relative x(n), y(n) racchiuse tra parentesi e separate dalla virgola).

Ad esempio per n=5, alla quinta riga, partendo dalla prima colonna, verrà scritto: E(.....) in cui al posto dei punti di sospensione compariranno le coordinate del quinto punto inserito dall'utente.

Facendo girare il programma con RUN si ottiene:



Alla fine dell'iterazione, una volta inserite le coordinate di tutti i punti il messaggio che compare in basso nella finestra non serve più, quindi va cancellato.

Con questa linea (1114) il programma viene rimandato alla subroutine che inizia alla linea 1750 in cui si cancellano le righe ormai inutili con le istruzioni per l'inserimento.

```
1114>GO SUB 1750
```

È conveniente usare un sottoprogramma in quanto si potrebbe aver bisogno di cancellare scritte nella finestra anche in altre parti del programma.

Le righe contenenti il messaggio vengono cancellate con un'iterazione che sovrainprime alle scritte di ciascuna riga 12 spazi bianchi.

```
1750>REM **cancella ultime righe
      parte bassa del video**
1760 FOR n=13 TO 21
1770 PRINT AT n,0;"
1780 NEXT n
1790 RETURN
```

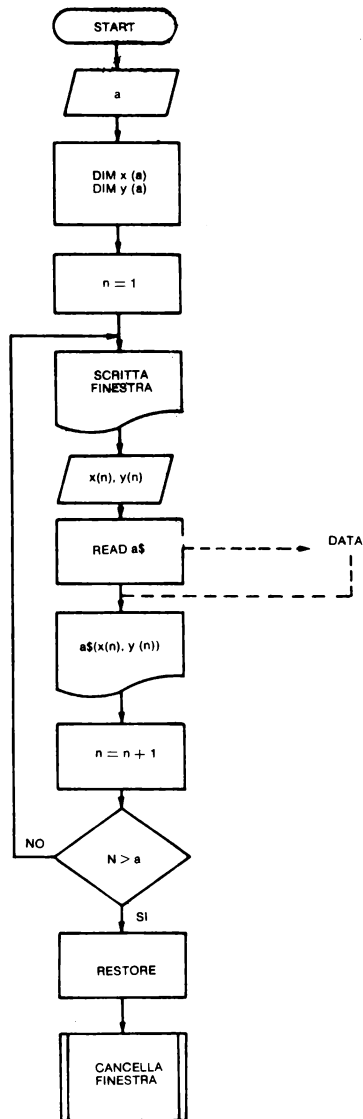
L'iterazione fa assumere ad n valori compresi fra 13 e 21 che corrispondono ai numeri delle righe su cui compare la scritta. La linea 1170 fa stampare alla riga n, partendo dall'inizio 0, dodici spazi bianchi.

La linea 1790 rimanda al programma principale.

È opportuno aggiungere un'ulteriore linea al programma che riporti il puntatore dei DATA alla posizione originale, operazione possibile con l'istruzione RESTORE.

```
1111>RESTORE
```

Il diagramma di flusso che sintetizza questa parte del programma è il seguente:



Esercizi

3. È possibile inserire un numero illimitato di punti?
R. In linea teorica sì, in pratica però il disegno sarà poco chiaro e dopo 12 punti si rischia lo scroll del video in quanto il messaggio inizia alla 13° riga.
4. Come va modificato il programma per cancellare tutta la finestra?
R. 1760 FOR n = 0 TO 21

2.6.3 Disegno della figura

Questa parte del programma può ulteriormente essere suddivisa:

- disegno della figura sul piano cartesiano;
- sistemazione a fianco dei vertici della figura delle lettere relative.

La figura viene disegnata tracciando una serie di segmenti che congiungono il primo vertice con il secondo, il secondo col terzo e così via: è necessario quindi portarsi al primo vertice della figura per poi incominciare da lì a tracciare la serie di lati consecutivi:

```
1115>REM **disegno figura**  
1120 PLOT x(1)+176,y(1)+88
```

La linea 1120 disegna il primo vertice della figura: le coordinate inserite dall'operatore sono i valori $x(1)$, $y(1)$ che l'elaboratore trova nelle DIM.

La correzione di +176 per l'ascissa e +88 per l'ordinata è dovuta alla differenza esistente tra il sistema di riferimento disegnato sul video e quello esistente nel calcolatore (Vedi MAT 8/a).

Con un'iterazione disegno ora i lati della figura, tutti fuorché quello che congiunge l'ultimo vertice con il primo, per cui verranno date istruzioni a parte.

In totale quindi i lati saranno $a-1$:

```
1130>FOR i=1 TO a-1  
1150 DRAW x(i+1)-x(i),y(i+1)-y(i)  
1160 NEXT i
```

È importante analizzare attentamente la linea:

```
1150>DRAW x(i+1)-x(i),y(i+1)-y(i)
```

L'istruzione `DRAW x,y` disegna un segmento che, partendo dall'ultima posizione di `PLOT` (x_0 , y_0), termina al pixel (x_0+x), (y_0+y).

Per definire l'incremento relativo delle coordinate dei vertici in successione rispettivamente per l'ascissa e per l'ordinata, è necessario eseguire per ogni segmento, la differenza tra la coordinata del vertice successivo ($i+1$) e quello di partenza (i).

Quindi l'incremento dell'ascissa del punto terminale di ogni segmento sarà:

$$x(i+1)-x(i)$$

ed egualmente per l'ordinata

$$y(i+1)-y(i)$$

Utilizzando questo sistema rimane escluso però il segmento che congiunge l'ultimo vertice della figura con il primo; per tracciarlo serve l'istruzione

```
1170>DRAW x(1)-x(a),y(1)-y(a)
```

Il sistema usato è analogo al precedente, solo che la differenza tra le coordinate viene effettuata tra i valori del primo vertice $x(1)$, $y(1)$ e quelli dell'ultimo $x(a)$, $y(a)$.

Con una nuova iterazione, a questo punto del programma accanto ai vertici della figura vengono scritte le lettere dell'alfabeto corrispondenti: queste ultime vengono prese nuovamente dai DATA della linea 1700 per mezzo dell'istruzione `READ a$`.

Il programma che ne risulta è il seguente:

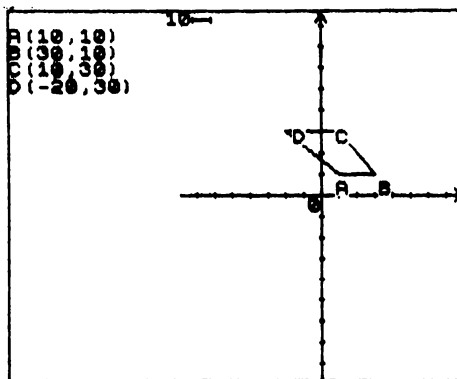
```

1180>FOR i=1 TO a
1190 READ a$
1200>PRINT AT INT (y(i)-88)/8,IN
T (x(i)+176)/8;a$
1205 NEXT i

```

La linea 1200 definisce, secondo la pagina PRINT, la posizione della lettera (vedi UDS MAT 8/a).

Facendo girare il programma con RUN si ottiene:



2.7 Traslazione di figure sul piano cartesiano

Servizio U.D.S.

CODICE MAT 12e

AUTORE PAOLA CATALANI

ANNO 1984

SCUOLA MEDIA STATALE "DI VITTORIO" - Pioltello (MI)

TITOLO "Traslazione di figure sul piano cartesiano"

RIASSUNTO

Viene disegnata la figura traslata una volta inserite le equazioni di traslazione. L'operatore deve poi inserire le coordinate dei vertici della figura traslata che verranno confrontate con quelle in memoria dell'elaboratore.

Curatori: PIERO D'ALFONSO - GIOVANNI MARTORANA

Elementare	<input type="checkbox"/>	Alfabetizzazione informatica	<input type="checkbox"/>
Media	<input checked="" type="checkbox"/>	linguistica	<input type="checkbox"/>
Media superiore	<input type="checkbox"/>	scienze / matematica	<input checked="" type="checkbox"/>
		tecnica	<input type="checkbox"/>
		grafica	<input type="checkbox"/>
BASIC	<input checked="" type="checkbox"/>	LOGO	<input type="checkbox"/>
		PASCAL	<input type="checkbox"/>
iterazione FOR/NEXT	<input checked="" type="checkbox"/>	(solo per BASIC)	
scelta condizionata IF/THEN	<input checked="" type="checkbox"/>		
matrici di memoria DIM	<input checked="" type="checkbox"/>		
READ/DATA	<input checked="" type="checkbox"/>		
DATA su FILE	<input type="checkbox"/>		
soubroutine GOSUB/RETURN	<input checked="" type="checkbox"/>		
menù	<input type="checkbox"/>		
memoria di schermo	<input type="checkbox"/>		
word processing	<input type="checkbox"/>		
collegamenti a rete	<input type="checkbox"/>		

SINCLAIR ZX 81	<input type="checkbox"/>	
SINCLAIR SPECTRUM	<input checked="" type="checkbox"/>	MINIDRIVE <input checked="" type="checkbox"/>
SINCLAIR QL	<input type="checkbox"/>	
COMMODORE VIC 20	<input type="checkbox"/>	DRIVE PER FLOPPY <input type="checkbox"/>
COMMODORE C 64	<input type="checkbox"/>	DRIVE PER FLOPPY <input type="checkbox"/>
M 10 OLIVETTI	<input type="checkbox"/>	
M 20 OLIVETTI	<input type="checkbox"/>	DRIVE <input type="checkbox"/>
SHARP MZ 700	<input type="checkbox"/>	DRIVE <input type="checkbox"/>
APPLE II	<input type="checkbox"/>	DRIVE PER FLOPPY <input type="checkbox"/>
I.B.M. PC	<input type="checkbox"/>	DRIVE <input type="checkbox"/>

Progetto "Banca delle Unità Didattiche"

Per ottenere la traslazione di una figura disegnata sul piano cartesiano è necessario proseguire il programma impostato nella UDS MAT 11/d.

Innanzitutto l'operatore deve inserire le equazioni di traslazione per mezzo di un INPUT.

```
1300>REM **traslazione**
1305 PRINT AT 17,0;"Inserisci le
": PRINT AT 19,0;"equazioni di":
PRINT AT 21,0;"traslazione":
1310 INPUT "x' = x"; c; "y' = y"; d
```

Con la linea 1305 si utilizza la finestra presente sulla parte sinistra dello schermo, per informare l'utente di ciò che l'elaboratore attende.

Il messaggio è scritto rispettivamente sulla 17° e 21° riga in modo tale che non sia costituito da più di 12 caratteri per non uscire dalla finestra.

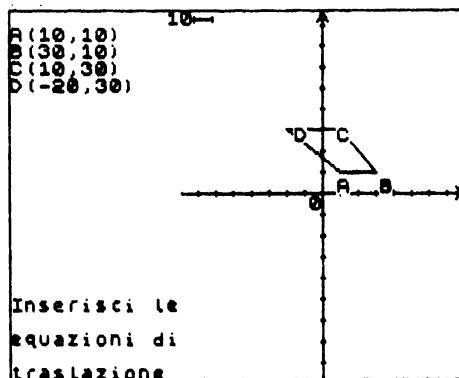
La linea 1340 di INPUT scrive nella parte bassa l'impostazione dell'equazione.

$$x' = x$$

l'operatore dovrà completarla inserendo la componente del vettore di traslazione lungo l'asse x: questo valore verrà conservato nell'indirizzo di memoria c.

Lo stesso procedimento avviene per l'altra equazione (componente lungo l'asse y) per la quale il valore inserito viene indirizzato nella memoria d.

Facendo girare il programma con RUN si ottiene:



A questo punto la scritta contenuta nella parte bassa della finestra non è più necessaria e deve essere cancellata.

```
1315>GO SUB 1750
```

La linea 1315 rimanda alla subroutine con cui viene cancellata la parte inferiore della finestra:
linee 1750 - 1790 UDS MAT 11/d

A questo punto è necessario che l'elaboratore:

- 1) calcoli i nuovi valori delle coordinate dei vertici della figura traslata;
- 2) disegni la nuova figura;
- 3) scriva le lettere dell'alfabeto accanto ai vertici della figura traslata in modo che corrispondano a quelli della figura di partenza.

2.7.1 Calcolo delle coordinate per punti traslati

Vengono aperte e dimensionate due memorie che contengono i nuovi valori delle coordinate:

```
1317>REM calcolo coordinate punt  
i traslati  
1320 DIM v(a): DIM z(a)
```

Nell'indirizzo di memoria v verranno sistemate le ascisse degli a punti traslati, nell'indirizzo di memoria z le rispettive ordinate.

In modo automatico verranno calcolati i nuovi valori delle coordinate

```
1330>FOR i=1 TO a  
1340 LET v(i)=x(i)+c  
1350 LET z(i)=y(i)+d  
1360>NEXT i
```

L'iterazione compresa tra le linee 1330 e 1360 modifica i valori delle coordinate di ogni punto della figura.

La linea 1340 cambia il valore dell'ascissa:

$LET\ v(i) = x(i) + c$

la nuova ascissa $v(i)$ sarà uguale alla corrispondente $x(i)$ della figura originaria, relativa cioè allo stesso punto, incrementata però nel valore c .

Va ricordato che c è la componente del vettore di traslazione lungo l'asse x .

La linea 1350 effettua la medesima operazione per i valori dell'ordinata.

A questo punto del programma dunque in

DIM x(a) sono contenuti i valori delle ascisse dei vertici della figura di partenza

DIM y(a) sono contenuti i valori delle ordinate dei vertici della figura di partenza con lo stesso ordine (A, B, C...)

DIM v(a) sono contenuti i valori delle ascisse dei vertici corrispondenti della figura traslata (A', B', C'....)

DIM z(a) sono contenuti i valori delle ordinate dei vertici corrispondenti della figura traslata

2.7.2 Disegno della figura traslata

Con un procedimento analogo a quello usato in UDS MAT 11/d per disegnare la figura di partenza, viene disegnata quella traslata.

```
1365>REM disegno figura traslata  
1370 PLOT v(1)+170, z(1)+88  
1380 FOR i=1 TO a-1  
1390 DRAW v(i+1)-v(i), z(i+1)-z(i)  
1400 NEXT i  
1410 DRAW v(1)-v(a), z(1)-z(a)
```

La linea 1370 fissa il primo vertice della figura traslata, l'iterazione delle linee 1380-1390-1400 traccia i lati della figura fuorché l'ultimo, per il quale è necessaria l'istruzione della linea 1410.

2.7.3 Scrittura delle lettere ai vertici

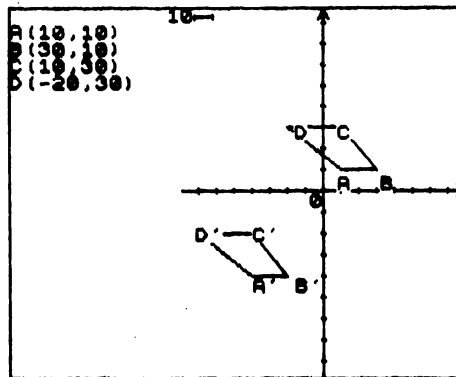
È necessario utilizzare ancora i DATA, quindi si dovrà rimettere il puntatore alla posizione di partenza con l'istruzione

1415>RESTORE

A questo punto viene nuovamente usato il procedimento spiegato nella UDS MAT 11/d per collocare accanto ai vertici corrispondenti la stessa lettera dell'alfabeto con a fianco il segno "".

```
1420>FOR i=1 TO a
1430>READ a$
1440 PRINT AT INT (Z(i)-88)/8,IN
T (v(i)+175)/8;a$;" "
1450 NEXT i
```

Facendo girare il programma con RUN si ottiene:



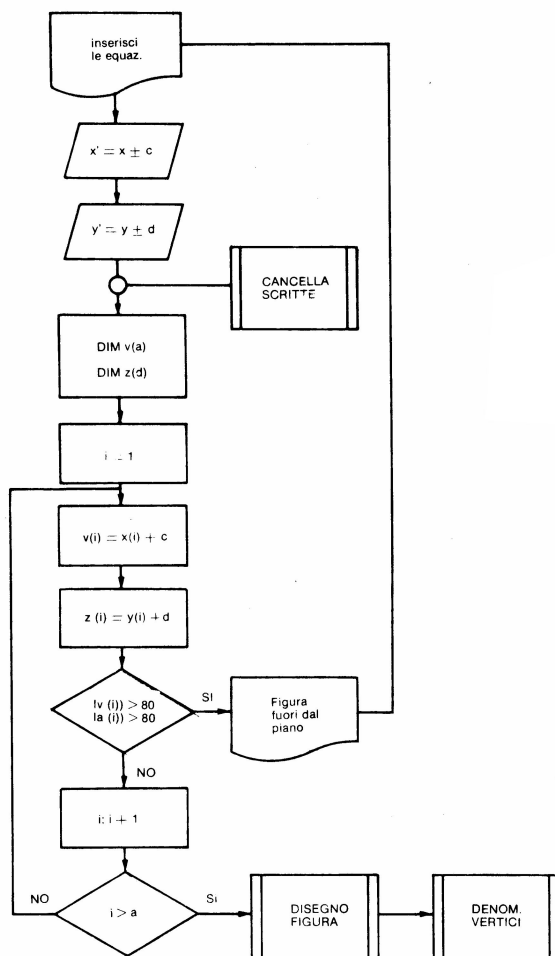
È necessario prevedere il caso in cui l'operatore inserisca equazioni di traslazione tali da far uscire la figura traslata dal piano cartesiano tracciato sullo schermo: in questo caso il programma si ferma segnalando errore B, integer out of range.

Per evitare questo inconveniente è sufficiente aggiungere:

```
1355>IF v(i)>88 OR v(i)<-88 OR Z
(i)>88 OR Z(i)<-88 THEN PRINT AT
13,0;"Figura fuori": PRINT AT 1
5,0;"dal piano": GO TO 1305
```

Nel caso in cui una qualsiasi delle nuove coordinate esca dal piano, compare nella finestra, alle righe 13 e 15, il messaggio "Figura fuori dal piano" e si viene rimandati alla linea 1305 in cui verranno inserite nuove equazioni di traslazione.

Il diagramma di flusso del programma così modificato è il seguente:



Il programma prosegue ora con l'obiettivo di coinvolgere l'operatore nella comprensione dei meccanismi di traslazione, chiedendo all'utente di inserire le coordinate dei vertici della figura traslata.

```

1500>REM **calcolo coordinate de
i punti traslati**
1520 RESTORE
1522 PRINT AT 13,0;"Scrivi le":
PRINT AT 15,0;"coordinate"
1524 PRINT AT 17,0;"del punto":
PRINT AT 19,0;"traslato"

```

La linea 1520 rimette il puntatore dei DATA nella posizione di partenza in quanto si dovranno utilizzare nuovamente le lettere dell'alfabeto.

Nella finestra compare il messaggio per l'operatore.

Ancora una volta le coordinate inserite dovranno essere indirizzate in una dimensione di memoria:

```

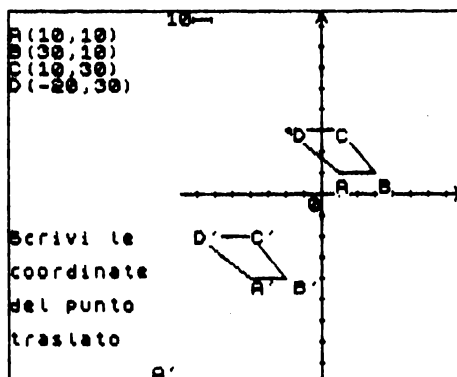
1575>DIM e(a): DIM f(a)

```

Nella DIM e(a) saranno inserite le ascisse, nella DIM f(a) le ordinate dei vertici della figura traslata, quindi in

DIM x(a) e DIM y(a) si trovano rispettivamente le ascisse e le ordinate della figura di partenza
 DIM v(a) e DIM z(a) rispettivamente le ascisse e le ordinate dei vertici della figura traslata in base alle equazioni inserite dall'utente
 DIM e(a) e DIM f(a) rispettivamente le ascisse e le ordinate inserite dall'utente che verranno confrontate con quelle contenute in DIM v(a) e DIM z(a)

Facendo girare il programma con RUN, si ottiene



Con un'iterazione vengono ora inserite le coordinate dei punti nell'ordine

```
1580>FOR i=1 TO a
1590 READ a$
1593 PRINT AT 21,10;a$;"
1610>INPUT e(i);f(i)
1620 IF e(i)=v(i) AND f(i)=z(i)
THEN GO TO 1640
```

L'istruzione della linea 1580 fa partire l'iterazione che si ripeterà per ognuno degli a vertici della figura.

Con la linea 1590 l'elaboratore legge nei DATA la lettera dell'alfabeto corrispondente al vertice di cui l'utente deve inserire le coordinate: la linea 1593 scrive la lettera con a fianco l'apice alla 21° riga della finestra (l'operatore cioè sa di quale punto gli vengono richieste le coordinate).

Le istruzioni della linea 1610 servono per introdurre i valori delle coordinate che verranno indirizzati nelle rispettive dimensioni di memoria.

La linea 1620 effettua il confronto tra i valori e(i) ed f(i) inseriti dall'utente con i rispettivi valori v(i) e z(i) che l'elaboratore aveva calcolato automaticamente alle linee 1340 e 1350: con e(i)=v(i) si confrontano le ascisse, con f(i)=z(i) le ordinate.

Nel caso in cui i valori inseriti dall'operatore e quelli calcolati dall'elaboratore coincidessero l'ultima istruzione della linea 1620 fa fare un salto fino alla linea 1640 (GO TO 1640) dove si trova l'istruzione:

```
1640>PRINT AT (a+i+2),0;a$;" ("
e(i);",",f(i);")"
```

Queste istruzioni fa scrivere nella finestra la lettera dell'alfabeto a\$ con a fianco l'apice "" e di seguito, fra parentesi, le coordinate esatte del punto.

La riga su cui scrive viene determinata dall'espressione a+i+2, le coordinate dei vertici della figura traslata vengono scritte due righe sotto quelle degli a vertici della figura di partenza (a+2), uno sotto l'altro nell'ordine i dell'iterazione.

Con l'istruzione della linea successiva si chiude l'iterazione.

```
1650>NEXT i
```

I valori inseriti dall'operatore però possono essere errati, in questo caso il programma non va alla linea 1640 ma procede con le istruzioni:

```
1625>GO SUB 1750
1628 GO SUB 1800
```

GOSUB 1750 rimanda al sotto programma che cancella le ultime righe della finestra.
GOSUB 1800 manda invece ad una subroutine che scrive in basso sulla finestra le equazioni:

```
1795>REM **scrive equazioni**
1800 PRINT AT 17,0; "Le equazioni"
1810 PRINT AT 18,0; "sono"
1810 IF c>0 THEN PRINT AT 19,0;
"x' = x + c; GO TO 1830
1820 PRINT AT 19,0; "x' = x"; c
1830 IF d>0 THEN PRINT AT 20,0;
"y' = y + d; GO TO 1850
1840 PRINT AT 20,0; "y' = y"; d
1850 RETURN
```

Con l'istruzione di linea 1800 viene scritto sulla 17° e 18° riga: "Le equazioni sono".

Le altre 4 linee servono per scrivere correttamente le equazioni, secondo un procedimento già spiegato nella UDS MAT 10/c.

Questa ulteriore informazione fornita dal programma aiuta l'operatore al momento dell'inserimento delle coordinate.

A questo punto un'istruzione 1630 GO TO 1610 servirebbe a ricominciare da capo l'inserimento per quel punto: però con una tale istruzione, in caso di continuo inserimento di valori errati da parte dell'utente, questa parte di programma continuerebbe a girare all'infinito.

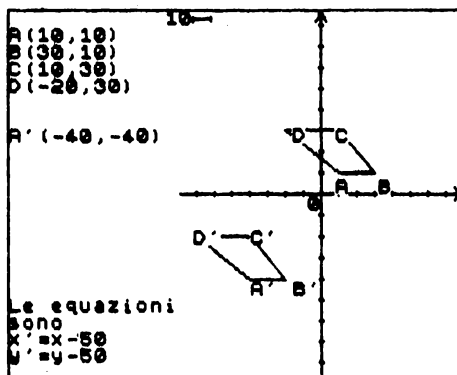
È necessario prevedere quindi un numero massimo di inserimento, al di là del quale l'elaboratore dia automaticamente la risposta corretta: per far ciò è necessario introdurre una nuova iterazione:

```
1595>FOR m=1 TO 3
1629>IF m=3 THEN LET e(i)=v(i):
LET f(i)=z(i): GO TO 1620
1630>NEXT m
```

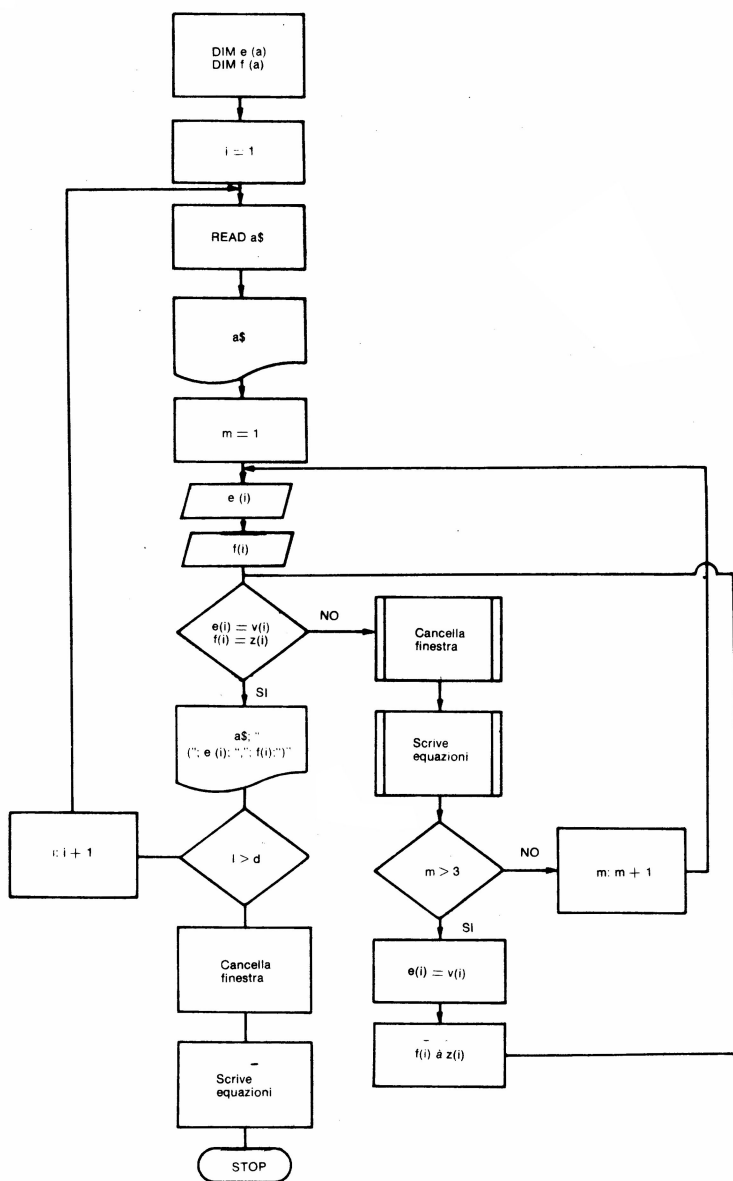
Questa iterazione permette all'operatore di inserire per tre volte valori errati per ogni punto, dopo di che la linea 1629 fa assumere ai valori inseriti $e(i)$ ed $f(i)$ quelli corretti contenuti negli indirizzi di memoria $v(i)$ e $z(i)$ quindi il programma viene fatto proseguire rimandandolo alla linea 1620 dove si effettua il confronto che naturalmente questa volta avrà esito positivo.

Ricapitolando quindi in caso di errore è possibile inserire per tre volte i valori per ogni vertice, al primo errore in basso nella finestra compaiono le equazioni di traslazione per aiutare l'operatore nell'inserimento.

Facendo girare il programma con RUN si ottiene:



Il diagramma di flusso che sintetizza questa parte di programma è il seguente:



Rimane da analizzare ora l'ultima parte del programma, una volta cioè inserite le coordinate di tutti i vertici della figura.

```

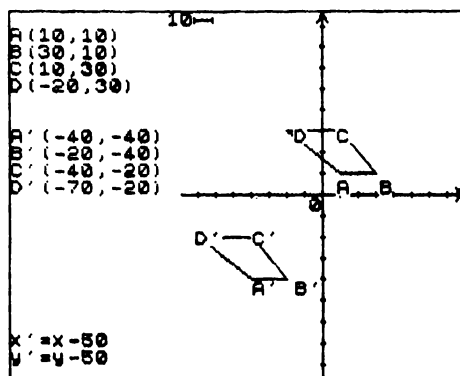
1655 > GO SUB 1750
1660 GO SUB 1810
1660 RESTORE
  
```

La parte bassa della finestra viene cancellata con l'istruzione di linea 1655.

Successivamente (linea 1658) si rimanda alla subroutine che scrive in basso nella finestra le equazioni, questa volta però il programma partirà dalla linea 1810 e non più dalla 1800 come in precedenza: viene tralasciata in questo caso la scritta "Le equazioni sono" per non togliere spazio alla scrittura delle coordinate dei vertici nel caso di poligoni con numerosi vertici.

L'istruzione successiva: 1660 RESTORE rimette alla posizione iniziale il puntatore dei DATA in modo che il programma possa essere fatto girare nuovamente.

La situazione sul video a questo punto è la seguente



Con una serie di istruzioni si può prevedere il sistema per far ricominciare il programma dal principio.

```
1687>PRINT AT 21,0:"Premi un tes
to per ricominciare"
1690 IF INKEY$("<>") THEN GO TO 16
90
1693 IF INKEY$="" THEN GO TO 169
3
1695 IF INKEY$("<>") THEN CLS : GO
TO 1
```

L'istruzione della linea 1695 contiene la funzione INKEY\$ che legge la tastiera, basta cioè premere un tasto qualsiasi, diverso da niente (<> "") per cancellare lo schermo (CLS) e ricominciare da capo (GOTO 1).

Esercizi

— Come si deve modificare il programma per far copiare dalla stampante l'immagine del video alla fine del programma?

R.

```
1665>INPUT "Hai la stampante? (s
/n)"; b$
1670 IF b$="n" THEN GO TO 1687
1675 INPUT "Vuoi copiare il dise
gno? "; b$
1680 IF b$="s" THEN COPY
```

— come si deve modificare il programma per consentire 2 possibilità di errore nell'inserimento anziché 3?

R.

```
1695>FOR n=1 TO 2
1699>IF n=2 THEN LET e(i)=v(i):
LET f(i)=z(i): GO TO 1620
```

— Qual'è il valore massimo della variabile a (numero dei vertici) tale da fare in modo che sulla finestra compaiono le coordinate dei vertici delle due figure e le equazioni senza che si sovrappongano le scritte?

R 8: vertici prima figura	righe 1 → 8
vertici seconda figura	righe 10 → 17
equazioni	righe 19 → 20

Esempi di software didattico

3.1 La rappresentazione di tessuti: una simulazione per l'educazione tecnica

SALVATORE FRANCESCO STILLO

3.1.1 Che cosa è la tessitura

Il passaggio obbligato per chiunque debba affrontare il tema di cosa significa analizzare un tessuto è necessariamente *l'intreccio* caratterizzabile come una particolare forma di incastro tra due ordini di fili (uno chiamato ordito e l'altro trama) in cui:

- i fili di uno stesso ordine sono tra loro paralleli;
- ogni filo è perpendicolare a quelli dell'altro ordine.

In altri termini l'intreccio ipotizzato coprirà un campo con un reticolo a maglie quadrate.

La relazione di intreccio è riducibile a due stati elementari: sotto-sopra, in accordo con il fatto che preso un qualunque filo di uno dei due ordini, ogni filo dell'altro ordine, nell'incrociarsi con esso gli passerà o sotto o sopra.

Ovviamente gli stati sotto-sopra non assicurano da soli il realizzarsi dell'intreccio, perché ciò si verifichi è necessario che si predetermini per ogni filo di trama e d'ordito un continuo passaggio tra sopra e sotto e viceversa.

Stanti queste condizioni è facile rendersi conto che quando un filo di trama va ad incrociare l'insieme di fili di ordito per realizzare l'intreccio è molto comodo che scorra sopra quelli che gli dovranno stare sotto essendo stati sollevati tutti quelli che gli dovranno stare sopra.

Questa soluzione definisce dei requisiti necessari per la macchina che ha il compito di realizzare l'intreccio: il telaio.

Il telaio, infatti, oltre a curare il mantenimento del parallelismo per fili di uno stesso ordine e la complanarità del complesso di fili, dovrà avere cure differenti per l'ordito e per la trama.

Per l'ordito curerà in particolare:

- il mantenimento dell'ordine con il quale i fili si succedono;
- il sollevamento di parte di essi all'atto dell'incrocio.

Per la trama assicurerà:

- l'attraversamento dell'ordito nel rispetto dello stato associato ad ogni incrocio.

La modalità classica di funzionamento dei telai consiste nel far attraversare ad ogni filo d'ordito l'occhiello di un elemento libero di muoversi alternativamente verso il basso e verso l'alto lungo un piano ortogonale a quello su cui è adagiato l'ordito stesso.

Il pezzo di macchina capace di sollevare in questo modo o di abbassare un singolo filo di ordito prende il nome di "maglia".

Le maglie possono essere unite e formate una struttura che le vincola a compiere simultaneamente lo stesso movimento verso l'alto o verso il basso.

Una tale struttura prende il nome di liccio.

Utilizzando più licci saremo in grado di compiere una vasta gamma di intrecci.

Il filo di trama, sia che scorra avanti o indietro, potrà quindi attraversare l'ordito passando in una sorta di galleria (passo d'ordito) che viene a formarsi non appena un liccio solleva alcuni fili traslando verso l'alto.

Ammesso di avere a disposizione i fili di ordito già predisposti all'intreccio se consideriamo una singola riga di trama si potrà dire, per ogni incrocio con un filo di ordito, se la trama gli passa sotto o sopra.

Ciò si traduce con numeri binari (1 e 0) oppure con due colori (bianco e nero).

Seguiamo una parte di ragionamento attenendoci alla descrizione con cifre binarie.

Ad ogni riga corrisponderà un numero binario formato da tanti uni e zeri quanti sono gli incroci.

Se sulla riga si nota una ripetizione ciclica delle cifre, è plausibile conservare "l'elemento di cifre minimale" come modulo riga.

Esaminiamo allora le più semplici combinazioni modulari:

- A. 1° passaggio: 1 0
2° passaggio: 0 1

- B. 1° passaggio: 0 0 1
2° passaggio: 0 1 0
3° passaggio: 1 0 0

In A le combinazioni (00) e (11) sono da scartare poiché escludono l'intreccio, in altre parole in ogni modulo deve esistere almeno uno "0" o un "1".

Analogamente in B sono da escludere (000) e (111).

Le coppie di cifre di A producono comunque, fatto salvo l'inizio di riga, lo stesso intreccio di riga, cioè:

1010101010101010 ecc.

Lo stesso dicasi per B dove le tre terne realizzano la riga

001001001001001 ecc.

Tutto ciò vale per le righe prese singolarmente, ma noi sappiamo che lo scopo è quello di produrre superfici, saremo quindi interessati a considerare l'accoppiamento di righe.

Se le righe sono accoppiate non è più vero che la combinazione A 1° è la stessa di A 2°; infatti ora la cifra d'inizio è determinante tanto che, tenendone conto si ottiene:

- A. 1. 1010101010101010
2. 0101010101010101 caso vero

invece di

1010101010101010 .
1010101010101010 caso falso

Solo nel caso vero ho realizzato un intreccio.

Allo stesso modo per le combinazioni B 1°, B 2°, B 3° si ottiene:

- B. 1. 001001001001001001
2. 010010010010010010 caso vero
3. 100100100100100100

ben diverso da

001001001001001001
001001001001001001
001001001001001001

caso falso

Di nuovo osserviamo che solo nel caso vero ho realizzato un intreccio.

Abbandoniamo ora le cifre binarie per affidare la rappresentazione dell'intreccio a un campo quadrettato dove al posto di "1" ci sarà un quadretto nero e a quello di "0" un quadretto bianco.

Fissiamo una convenzione che fa corrispondere a:

quadretto nero	filo d'ordito sopra filo di trama
quadretto bianco	filo d'ordito sotto filo di trama

A questa scrittura si dà il nome di messa in carta o grafico.

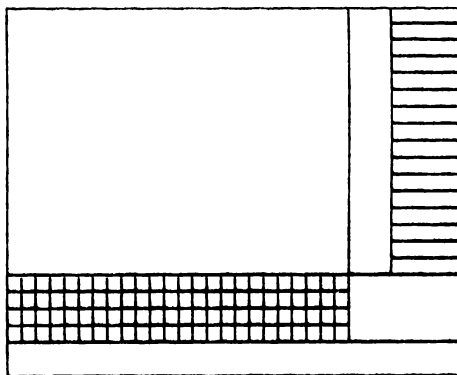
Di nuovo potremo parlare di modulo nel caso in cui si riscontrino delle ripetizioni cicliche, ma ora siamo in presenza di una superficie quadrettata a bianco e nero quindi si tratterà di isolare quel quadrato o rettangolo che sta nei confronti del tessuto nella stessa relazione che lega una mattonella al suo pavimento.

La messa in carta si configura come una scrittura simbolica dell'intreccio.

A prima vista si potrebbe essere però indotti ad assegnare a questa messa in carta un valore rappresentativo maggiore di quello che le spetta e cioè "vedere" in quella quadrettatura bianca e nera non tanto la stesura in simboli dell'intreccio quanto il "disegno" di un tessuto a due colori.

Ciò è vero nel caso che l'ordito sia nero e la trama sia bianca.

Quanto appena detto fa capire l'esigenza di disporre, accanto alla rappresentazione simbolica, di un reticolo a maglie quadrate nel quale poter scrivere il disegno del tessuto colorando secondo il colore del filo di ordito o di trama il quadratino corrispondente.



3.1.2 Spiegazione del programma

Che cosa fa il programma

Il programma "ates" (archivio tessuti) consente di visualizzare sullo schermo:

- la messa in carta (o grafico) delle armature;
- la visualizzazione del tessuto in colore.

Una stampante collegata al computer ne può riprodurre su carta le relative videate. Durante l'elaborazione si potrà osservare la simulazione dell'intreccio sia nella fase di scrittura della messa in carta che nel disegno del tessuto.

Nel primo caso nell'area di scrittura del tessuto si stampano caselle bianche o nere in relazione alla posizione reciproca dei fili all'atto del loro intreccio.

Mentre nel disegno del tessuto colorato le singole posizioni di stampa sono soggette al colore del filo in evidenza in quel punto.

Avvalendosi della possibilità di controllare la variabile colore, l'operatore può valutare, dalla visualizzazione di uno scorcio della pezza tessuta, l'effetto cromatico che l'accostamento dei colori produce sul tessuto stesso.

Viceversa facendo corrispondere una videata a ciascuna delle armature si potranno apprezzare gli effetti prodotti sul tessuto dai diversi tipi di intreccio.

Come si usa il programma

Se il programma viene caricato nella memoria RAM del computer da nastro magnetico, automaticamente appare il menù principale.

In tutti gli altri casi può essere mandato in esecuzione con il comando RUN o GOTO 10.

Nel menù sono elencate le 4 procedure ammesse: inserendo il numero corrispondente alla procedura scelta si entra nel sottoprogramma relativo.

Dopo un RUN o quando si vogliono cambiare i colori il primo sottoprogramma da lanciare è "per lettura dati e scelta colori" per inizializzare le memorie che saranno usate nel programma.

Il numero 1 attiva la procedura di visualizzazione delle messe in carta, controllata anch'essa da sottomenù.

Anche qui le procedure sono attivate inserendo il numero relativo a quella scelta.

Prerequisiti

Si danno per acquisite le seguenti abilità:

- saper gestire un programma da menù;
- conoscenza dell'uso delle funzioni BASIC VAL, INKEY\$, STR\$, CODE;
- conoscenza della tecnica TOP DOWN di risoluzione dei problemi;
- saper organizzare uno spazio di memoria a 3 dimensioni;
- capacità di gestione della grafica dello Spectrum e degli attributi delle posizioni di stampa.

Il programma principale

Come già detto, lo scopo di questo programma è quello di simulare l'intreccio a telaio visualizzando la messa in carta e il tessuto colorato delle armature archiviate.

La risoluzione del problema è basata sul controllo dei dati contenuti in due matrici di cui diremo più avanti; inoltre è espressamente previsto che l'operatore scelga il colore dei fili da intrecciare.

Quindi è giustificata la presenza di una procedura di lettura dati e scelta colori.

L'abbandono del programma è possibile solo attivando la relativa procedura.

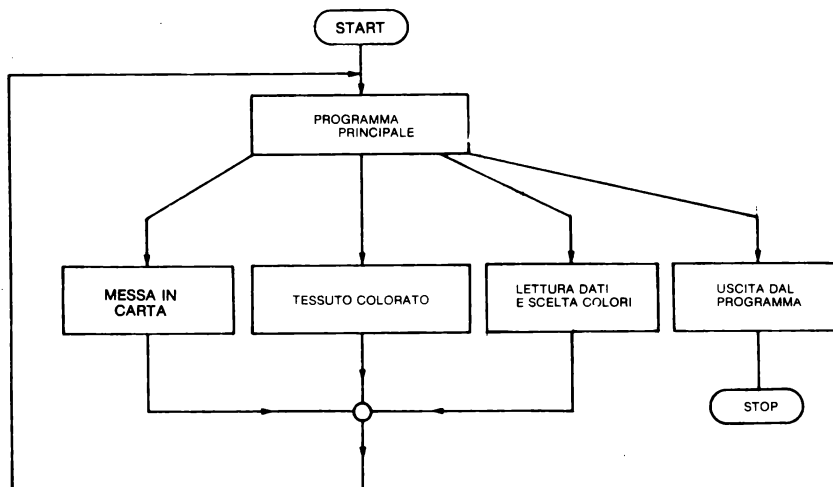
Riepilogando, il problema complessivo può essere scomposto nei seguenti sottoproblemi:

- a) messa in carta;
- b) tessuto colorato;
- c) lettura dati e scelta colore;
- d) abbandono del programma

Per ogni sottoproblema individuato creiamo un sottoprogramma.

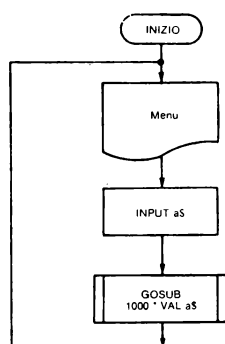
Ognuno di essi è un problema a sé che si ricollega agli altri attraverso un menù principale e racchiude in sé tutte le istruzioni necessarie per la risoluzione di uno dei problemi in cui è stato diviso il problema complessivo.

La struttura generale del programma è rappresentata nel seguente diagramma a blocchi.



Il programma principale, così detto perché non viene richiamato da altre parti del programma, comprende il modulo di controllo e il menù cioè la lista di problemi che il programma è in grado di risolvere e la indicazione delle operazioni da compiere per attivare le singole procedure.

Il diagramma a blocchi corrispondente al programma principale è riportato in figura:



La relativa codifica è data dalla lista di istruzioni di seguito riportata:

```

10 BRIGHT 0:CLS: REM RAPPRES
ENTRAZIONE DI TESSUTI
20 REM Programma principale
25 INK 0: BORDER 6
30 PRINT BRIGHT 1; FLASH 1; AT
4,12;"MENU";AT 6,10: "TESSITURA
40 PRINT BRIGHT 1;AT 9,4;"1 -p
er messa in carta
50 PRINT BRIGHT 1;AT 11,4;"2 -
per tessuto colorato"
60 PRINT BRIGHT 1;AT 13,4;"3 -
per lettura dati e
scelta colore
70 PRINT BRIGHT 1;AT 15,4;"4 -
per USCIRE dal progr"
90 PRINT BRIGHT 1;AT 20,1;"sel
ezione inserendo un numero
100 INPUT a$
110 IF a$="" THEN GO TO 100
120 IF CODE a$<49 OR CODE a$>52
THEN GO TO 100
130 GO SUB 1000*VAL a$
140 BRIGHT 0:CLS
150 GO TO 25

```

Le istruzioni:

```

100>INPUT a$
130>GO SUB 1000*VAL a$

```

controllano il rinvio ai singoli sottoprogrammi secondo la regola

1000 * valore matematico di a\$

Per evitare che eventuali errori di battitura interrompano l'esecuzione del programma sono state introdotte le linee:

```

110>IF a$="" THEN GO TO 100
120 IF CODE a$<49 OR CODE a$>52
THEN GO TO 100

```

con il compito di "filtrare" i caratteri in ingresso, lasciando passare solo le cifre comprese tra 1 e 5.

In figura è mostrata la videata corrispondente al menù principale prodotta dalle istruzioni comprese tra 10 e 90:

```

          MENU'
        TESSITURA

      1 -per messa in carta
      2 -per tessuto colorato
      3 -per lettura dati e
        scelta colore
      4 -per USCIRE dal progr

seleziona inserendo un numero

```

Le istruzioni 140 e 150 consentono di pulire lo schermo e fare apparire il menù al ritorno da un sottoprogramma.

Scomponiamo i sottoprogrammi secondo la tecnica top-down (dall'alto in basso, dal complesso a semplice) di risoluzione dei problemi fino ad arrivare a procedure direttamente codificabili secondo le strutture fondamentali del linguaggio basic.

Se si accede ai sottoprogrammi premendo 1 o 2 prima di essere passati dalla lettura dei dati e dalla scelta del colore, appena caricato il programma, questo non può girare perché privo delle necessarie istruzioni.

All'inizio dunque il primo sottoprogramma da scegliere è quello cui si accede premendo 3.

SUB Routine 3000 per lettura dati e scelta colore

La relazione di intreccio, come detto, è riducibile a due stati elementari: sotto-sopra.

Avvalendosi della convenzione di indicare con una casella nera (al posto della cifra 1) il passaggio della trama sotto l'ordito e con una casella bianca (al posto della cifra 0) il passaggio della trama sopra l'ordito, possiamo rappresentare anche graficamente l'intreccio (o armatura) per un dato rimettaggio.

Per il nostro lavoro fissiamo a sette il numero di armature da archiviare e a quattro il numero di licci di un ipotetico telaio che potrebbe intrecciare 16 fili di trama con 24 fili d'ordito.

Organizziamo uno spazio di memoria in cui memorizzare:

- a) il numero di messe in carta pari a 7;
- b) il numero di passaggi di trama nell'ordito pari a 16;
- c) il numero di fili di ordito pari a 24.

Poiché la posizione reciproca tra i fili di ordito e i fili di trama è funzione del tipo di armatura dobbiamo memorizzare i passaggi di trama nell'ordito per le sette armature.

È necessario allora dimensionare una matrice alfanumerica a tre indici: il primo numero controlla i tipi di armatura, il secondo la trama, il terzo l'ordito.

L'istruzione adatta allo scopo è:

```
3012>DIM t$(7,18,24)
```

Si noti che il secondo numero è 18 anziché 16: questo valore agevolerà una procedura di cui diremo più avanti.

Per ogni armatura possiamo allora memorizzare, in forma codificata (cifre 0 e 1), la posizione reciproca tra i fili di trama e di ordito all'atto del loro incrocio per tutti i passaggi di trama.

Accesso allo spazio di memoria per la tela

La tela è l'intreccio più semplice; il modulo di riga nei due sistemi di codifica è:

1 0


ripetendolo orizzontalmente 12 volte otteniamo:

1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0


che corrisponde al 1° passaggio di trama.

Poiché con l'intreccio si copre una superficie risulta conveniente individuare un modulo di estensione superficiale: il quadrato unitario.

01
10



simbolicamente

graficamente

ripetuto 12 volte orizzontalmente si ottengono i primi passaggi di trama:

010101010101010101010101 2° passaggio
101010101010101010101010 1° passaggio

I passaggi successivi si ripetono ciclicamente:
i dispari sono uguali al primo passaggio, i pari sono uguali al secondo passaggio.

Dimensioniamo una memoria ausiliaria di 15 elementi di 24 caratteri ciascuno per memorizzare i passaggi di trama nell'ordito con l'istruzione:

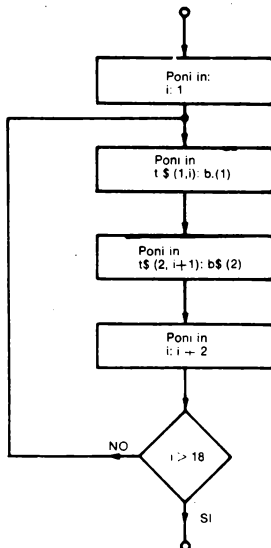
```
3013>DIM b$(15,24)
```

I primi due elementi li impiegheremo per i primi due passaggi di trama, gli altri saranno utilizzati successivamente.

```
3016>LET b$(1)="1010101010101010  
10101010"  
3017>LET b$(2)="0101010101010101  
01010101"
```

A questo punto siamo in grado di trasferire nella matrice t\$ tutte le informazioni necessarie a descrivere l'armatura tela.

La procedura risoltrice per il riempimento degli spazi di memoria della matrice t\$ destinati all'armatura tela è:



cui corrisponde la seguente codifica:

```
3100>REM Tela  
3110>FOR i=1 TO 18 STEP 2  
3120>LET t$(1,i)=b$(1): LET t$(1  
, i+1)=b$(2): NEXT i
```

Accesso alla memoria per il nattè

Diamo direttamente il modulo di superficie:
simbolicamente graficamente

1110
0001
0001



ripetuto orizzontalmente 6 volte ci dà i primi 3 passaggi di trama:

111011101110111011101110 3° passaggio
000100010001000100010001 2° passaggio
000100010001000100010001 1° passaggio



In questo caso sarà sufficiente memorizzare solo 2 delle 3 righe con le istruzioni

```
3023>LET b$(5) = "0001000100010001"
00010001
3025>LET b$(6) = "1110111011101110"
11101110
```

Il riempimento degli spazi di t\$(2,18,24) è ottenuto con le istruzioni:

```
3140>REM Nattè
3150 FOR i=1 TO 18 STEP 3
3160 LET t$(2,i)=b$(5): LET t$(2,i+1)=b$(5): LET t$(2,i+2)=b$(5)
NEXT i
```

Nell'istruzione 3150 troviamo uno STEP 3; nel caso in cui il valore finale di i fosse stato fissato in 16 con queste istruzioni avremo riempito gli spazi di memoria fino a 15° passaggio di trama escludendo il 16°.

Accesso allo spazio di memoria per il rep di ordito

Il rep di ordito è un'armatura derivata dalla tela. Il suo modulo è
simbolicamente graficamente

01
10
10



ripetuto orizzontalmente otteniamo i primi 3 passaggi di trama

010101010101010101010101 3° passaggio
101010101010101010101010 2° passaggio
101010101010101010101010 1° passaggio

Le prime due righe corrispondono al contenuto di b\$(1), la terza al contenuto di b\$(2); pertanto la codifica della procedura di riempimento degli spazi di memoria di t\$ è:

```
3200>REM Rep di ordito
3210 FOR i=1 TO 18 STEP 3
3220 LET t$(3,i)=b$(1): LET t$(3,i+1)=b$(1): LET t$(3,i+2)=b$(1)
NEXT i
```

Accesso alle memorie per il rep di trama

Il modulo di questa armatura è

simbolicamente

0011
1100

graficamente



Ripetuto orizzontalmente otteniamo i primi due passaggi di trama

001100110011001100110011
110011001100110011001100

che memorizziamo in b\$(3) e b\$(4) con le istruzioni

```
3010>LET b$(3) = "1100110011001100  
11001100  
3020 LET b$(4) = "0011001100110011  
00110011"
```

I passaggi successivi si ripetono ciclicamente per cui la codifica della procedura di riempimento degli spazi di memoria di t\$ è:

```
3250>REM rep di trama  
3260 FOR i=1 TO 18 STEP 2  
3270 LET t$(4,i) = b$(3): LET t$(4,  
i+1) = b$(4): NEXT i
```

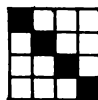
Accesso alla memoria per la saia

La saia è un'armatura fondamentale il cui modulo è

simbolicamente

1000
0100
0010
0001

graficamente



Ripetuto orizzontalmente 6 volte otteniamo i primi quattro passaggi di trama:

100010001000100010001000	4° passaggio
010001000100010001000100	3° passaggio
001000100010001000100010	2° passaggio
000100010001000100010001	1° passaggio

le cifre del primo passaggio di trama coincidono con il contenuto di b\$(5) pertanto memorizziamo solo i successivi 3 con le istruzioni

```
3025>LET b$(7) = "0010001000100010  
00100010  
3035 LET b$(6) = "0100010001000100  
01000100  
3050 LET b$(9) = "1000100010001000  
10001000"
```


Con le seguenti istruzioni trasferiamo il tutto negli spazi di t\$ destinati alla saia:

```
3300>REM Saia
3310 FOR i=1 TO 14 STEP 4
3320 LET t$(5,i)=b$(5): LET t$(5,i+1)=b$(7): LET t$(5,i+2)=b$(8)
: LET t$(5,i+3)=b$(9): NEXT i
```

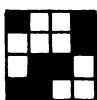
Accesso alla memoria per il BATAVIA

Il modulo di questa armatura è il seguente:

simbolicamente

1001
0011
0110
1100

graficamente



ripetuto sei volte orizzontalmente da:

100110011001100110011001	4° passaggio
001100110011001100110011	3° passaggio
011001100110011001100110	2° passaggio
110011001100110011001100	1° passaggio



Notiamo che il primo passaggio coincide con b\$ (3) e il terzo con b\$ (4) per cui memorizziamo il secondo e il quarto con le istruzioni:

```
3060>LET b$(10)="011001100110011
001100110"
3070>LET b$(11)="100110011001100
110011001"
```

Con le istruzioni seguenti memorizziamo negli spazi di t\$ (6, 18,24) questi valori ed estendiamo a tutti i passaggi di trama.

```
3350>REM Batavia
3360 FOR i=1 TO 14 STEP 4
3370 LET t$(6,i)=b$(3): LET t$(6,i+1)=b$(10): LET t$(6,i+2)=b$(4)
: LET t$(6,i+3)=b$(11): NEXT i
```

Accesso alla memoria per il raso

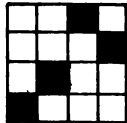
Il raso è una delle armature fondamentali.

Il suo modulo è:

simbolicamente

0010
0001
0100
1000

graficamente



ripetuto sei volte orizzontalmente dà:

001000100010001000100010	4° passaggio
000100010001000100010001	3° passaggio
010001000100010001000100	2° passaggio
100010001000100010001000	1° passaggio

Notiamo che il 1° passaggio è in b\$(9), il 2° in b\$(8), il 3° in b\$(5) e il 4° in b\$(7), pertanto facciamo riferimento al contenuto di questi elementi.

Le istruzioni per riempire gli spazi di t\$ (7,18,24) sono:

```
3400>REM Raso
3410 FOR i=1 TO 14 STEP 4
3420 LET t$(7,i)=b$(9): LET t$(7
,i+1)=b$(8): LET t$(7,i+2)=b$(5)
: LET t$(7,i+3)=b$(7): NEXT i
```

I colori

La volontà di visualizzare il tessuto colorato è legata alla possibilità di associare al singolo filo un codice di colore.

Dimensioniamo per questo una matrice alfanumerica a 2 numeri con l'istruzione:

```
3510>DIM c$(2,24)
```

in cui si hanno 2 elementi ognuno di 24 caratteri, il primo per l'ordito che usa tutti i caratteri a disposizione, il secondo per la trama che ne usa solo 16.

Per l'attribuzione del colore ai fili abbiamo previsto due possibilità:

- l'operatore desidera scegliere il colore dei singoli fili;
- l'operatore decide di assegnare un solo colore a tutto l'ordito e un solo colore a tutta la trama.

A questo scopo è preposto il messaggio:

```

Per un solo colore per l'ordito
e un solo colore per la trama
PREMI ENTER
altrimenti un qualunque tasto

```

ottenuto con la linea di programma 3520.

La linea di programma

```

3530>IF INKEY$="" THEN GO TO 353
0

```

invece determina un'attesa, finché non viene premuto un tasto.

Se il tasto premuto è ENTER (codice 13), cioè se la condizione logica CODE INKEY\$=13, descritta nella linea 3540, risulta verificata allora l'esecuzione prosegue con la procedura "un solo colore", altrimenti viene eseguita la procedura "vari colori".

Riportiamo il listato relativo a questa parte di programma

```

3500>REM Scelta colori
3510 DIM c$(2,24)
3520 PRINT BRIGHT 1;AT 8,0;"Per
un solo colore per l'ordito, e un
solo colore per la trama";TAB
8;FLASH 1;"PREMI ENTER";FLASH
0;"altrimenti un qualunque tasto
3530 IF INKEY$="" THEN GO TO 353
0
3540 IF CODE INKEY$=13 THEN GO T
0 3500

```

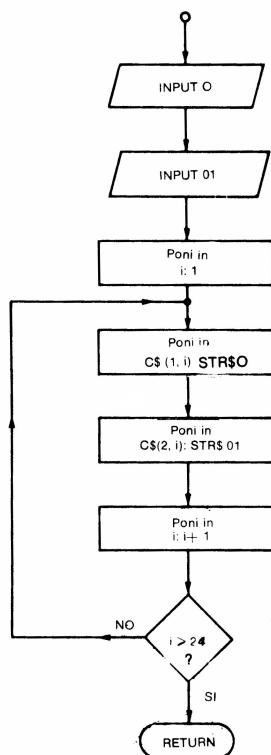
Procedura: un solo colore

In questa fase l'operatore sceglie il colore per l'ordito e il colore per la trama inserendo il relativo codice di colore in 0 e 01 rispettivamente.

Una iterazione, controllata della memoria i, assegna alla matrice bidimensionale c\$ i codici del colore.

La funzione STR\$ applicata alle memorie numeriche o e o1 (nota: o non 0) trasforma i rispettivi valori in cifre e ne consente l'assegnazione alla matrice alfanumerica c\$ per tutti i fili.

Il diagramma di flusso relativo è rappresentato in figura:



Mentre la codifica BASIC è mostrata nel seguente listato:

```

3800>CLS : REM Un solo colore
3810 INPUT "ORDITO, quale colore
?":o
3820 INPUT "TRAMA, quale colore?
?":o1
3830 FOR i=1 TO 24
3840 LET c$(1,i)=STR$ o
3850 LET c$(2,i)=STR$ o1
3860 NEXT i
3870 RETURN

```

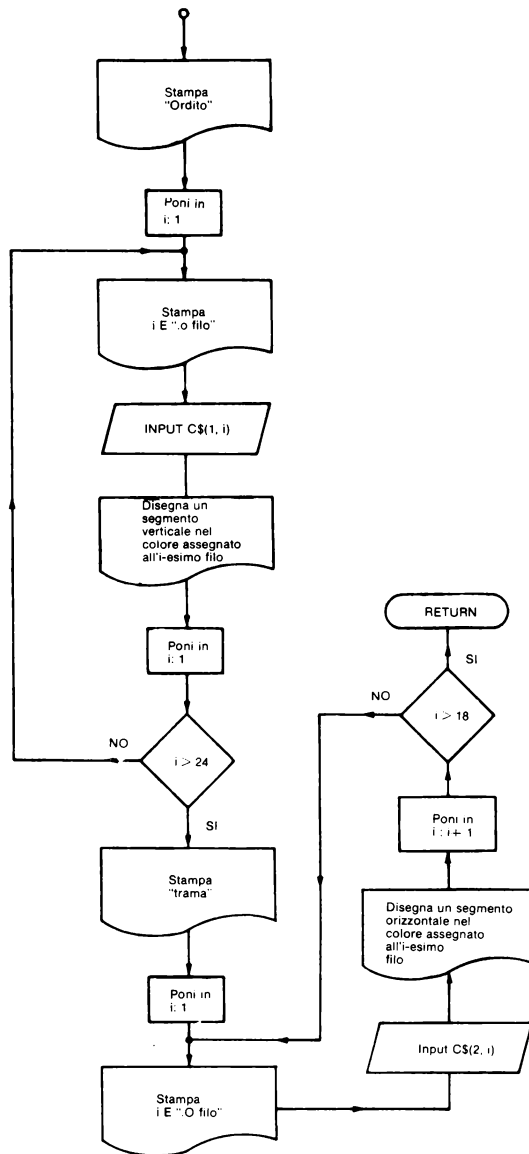
Procedura: vari colori

La scelta del colore è un passaggio obbligato per la progettazione del tessuto colorato.

La procedura che ci accingiamo a descrivere offre all'utente l'opportunità di scegliere il codice del colore da assegnare ai singoli fili, sia di trama che di ordito.

Per un controllo visivo del colore assunto dai fili disegneremo dei segmenti, orizzontali per la trama, verticali per l'ordito, nel colore del filo.

Il diagramma di flusso esemplifica in maniera efficace la risoluzione algoritmica di questa procedura:



Nel seguente listato sono riportate le linee di programma che codificano la procedura in linguaggio basic.

```

3600>REM Vari colori
3602 REM Scelta colore ordito
3605 BRIGHT 1: CLS : PRINT INK 0
: AT 1,0: "O R D I T O"
3610 FOR i=1 TO 24
3620 PRINT INK 0; AT 19,10; i; ".o
filo"
3630 INPUT "Quale colore? "; c$(1
,i)

```

```

3640 IF CODE c$(1,i)<48 OR CODE
c$(1,i)>57 THEN GO TO 3630
3650 PLOT 20+i*8,40: DRAW INK VA
L c$(1,i);0,100
3660 NEXT i
3665 PAUSE 50: CLS
3670 REM Scelta colore trama
3675 PRINT AT 1,11;"T R A M A"
3680 FOR i=1 TO 16
3690 PRINT INK 0;AT 20,12;i;"."
i=i+1
3700 INPUT "Quale colore? ";c$(2
i)
3705 IF CODE c$(2,i)<48 OR CODE
c$(2,i)>57 THEN GO TO 3700
3720 PLOT 72,20+i*8: DRAW INK VA
L c$(2,i);100,0
3730 NEXT i
3740 PAUSE 50: CLS : RETURN

```

La linea di programma 3650 consente di disegnare un segmento verticale nel colore del filo di ordito.

Quando il codice del colore del 24° filo è stato, inserito sullo schermo apparirà l'ordito completo: Mentre la linea 3720 mostrerà la trama.

In conclusione diamo il listato completo del sottoprogramma 3000.

```

3000>REM Lettura dati e scelta c
olori
3012 DIM t$(7,18,24)
3013 DIM b$(15,24)
3015 LET b$(1)="10101010101010
10101010"
3017 LET b$(2)="01010101010101
01010101"
3018 LET b$(3)="1100110011001100
11001100"
3020 LET b$(4)="0011001100110011
00110011"
3023 LET b$(5)="0001000100010001
00010001"
3025 LET b$(6)="1110111011101110
11101110"
3028 LET b$(7)="0010001000100010
00100010"
3035 LET b$(8)="0100010001000100
01000100"
3050 LET b$(9)="1000100010001000
10001000"
3060 LET b$(10)="011001100110011
00110011"
3070 LET b$(11)="100110011001100
11001100"
3100 REM Tela
3110 FOR i=1 TO 18 STEP 2
3120 LET t$(1,i)=b$(1): LET t$(1
,i+1)=b$(2): NEXT i
3140 REM Natta
3150 FOR i=1 TO 18 STEP 3
3160 LET t$(2,i)=b$(5): LET t$(2
,i+1)=b$(5): LET t$(2,i+2)=b$(6)
: NEXT i
3200 REM Rep di ordito
3210 FOR i=1 TO 18 STEP 3
3220 LET t$(3,i)=b$(1): LET t$(3
,i+1)=b$(1): LET t$(3,i+2)=b$(2)
: NEXT i
3250 REM rep di trama
3260 FOR i=1 TO 18 STEP 2
3270 LET t$(4,i)=b$(3): LET t$(4
,i+1)=b$(4): NEXT i
3300 REM Seta
3310 FOR i=1 TO 14 STEP 4
3320 LET t$(5,i)=b$(5): LET t$(5
,i+1)=b$(7): LET t$(5,i+2)=b$(8)
: LET t$(5,i+3)=b$(9): NEXT i
3350 REM Batavia

```

```

3360 FOR i=1 TO 14 STEP 4
3370 LET t$(6,i)=b$(3): LET t$(6,i+1)=b$(10): LET t$(6,i+2)=b$(4): LET t$(6,i+3)=b$(11): NEXT i
3400 REM Raso
3410 FOR i=1 TO 14 STEP 4
3420 LET t$(7,i)=b$(9): LET t$(7,i+1)=b$(8): LET t$(7,i+2)=b$(5): LET t$(7,i+3)=b$(7): NEXT i
3500 REM Scelta colori
3510 DIM c$(2,24)
3520 PRINT BRIGHT 1; AT 8,0; "Per un solo colore per l'ordito e un solo colore per la trama"; TAB 8; FLASH 1; "PREMI ENTER"; FLASH 0; "altrimenti un qualunque tasto"
3530 IF INKEY$="" THEN GO TO 3530
3540 IF CODE INKEY$=13 THEN GO TO 3600
3600 REM Vari colori
3602 REM Scelta colore ordito
3605 BRIGHT 1: CLS : PRINT INK 0; AT 1,0; "O R D I T O"
3610 FOR i=1 TO 24
3620 PRINT INK 0; AT 19,10;i; ".o filo"
3630 INPUT "Quale colore? "; c$(1,i)
3640 IF CODE c$(1,i)<48 OR CODE c$(1,i)>57 THEN GO TO 3630
3650 PLOT 20+i*8,40: DRAW INK VAL c$(1,i); 0,100
3660 NEXT i
3665 PAUSE 50: CLS
3670 REM Scelta colore trama
3675 PRINT AT 1,11; "T R A M A"
3680 FOR i=1 TO 16
3690 PRINT INK 0; AT 20,12;i; ".o filo"
3700 INPUT "Quale colore? "; c$(2,i)
3705 IF CODE c$(2,i)<48 OR CODE c$(2,i)>57 THEN GO TO 3700
3720 PLOT 72,20+i*8: DRAW INK VAL c$(2,i); 100,0
3730 NEXT i
3740 PAUSE 50: CLS : RETURN
3800 CLS : REM Un solo colore
3810 INPUT "ORDITO, quale colore? "; o
3820 INPUT "TRAMA, quale colore? "; t
3830 FOR i=1 TO 24
3840 LET c$(1,i)=STR$ o
3850 LET c$(2,i)=STR$ t
3860 NEXT i
3870 RETURN

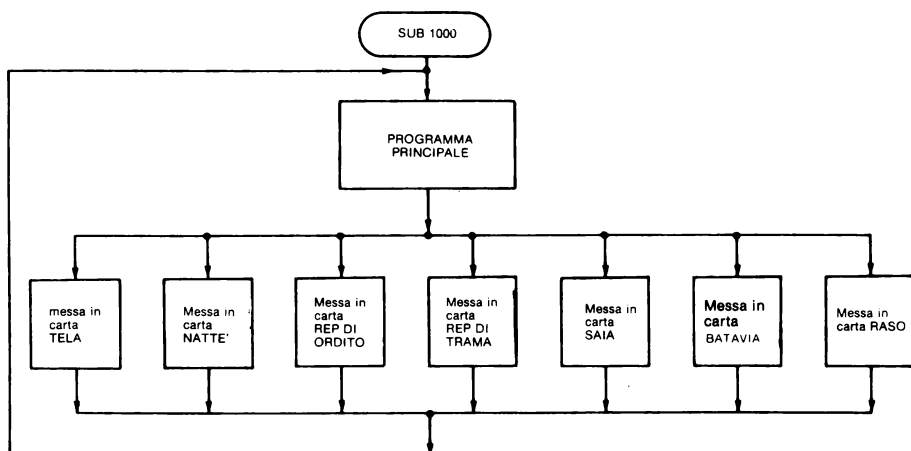
```

SUB Routine 1000 visualizzazione e stampa delle varie messe in carta

In base ai valori precedentemente definiti e archiviati possiamo visualizzare la messa in carta delle seguenti armature:

- tela,
- nattè,
- rep d'ordito,
- rep di trama,
- saia,
- batavia,
- raso.

A ciascuna delle messe in carta facciamo corrispondere un sottoprogramma che si lega agli altri con un programma principale (da non confondersi con quello del problema complessivo). Il problema "visualizzazione e stampa delle messe in carta" quindi possiamo esemplificarlo nel seguente diagramma a blocchi:



Occupiamoci della risoluzione algoritmica dei sottoproblemi di questa procedura.

Alla rappresentazione grafica dei tessuti, propriamente detta messa in carta o grafico, viene affidato il compito di comunicare:

- il tipo di intreccio;
- l'indicazione del numero progressivo associato ad ogni liccio utilizzato;
- l'ordine di passaggio dei fili di ordito negli occhielli delle maglie dei licci;
- il numero relativo al liccio (o numeri relativi ai licci) da azionare;
- il modulo.

Operiamo allora per pervenire ad uno schema standard su carta quadrettata del quale fare uso per rappresentare le varie armature complete delle informazioni elencate.

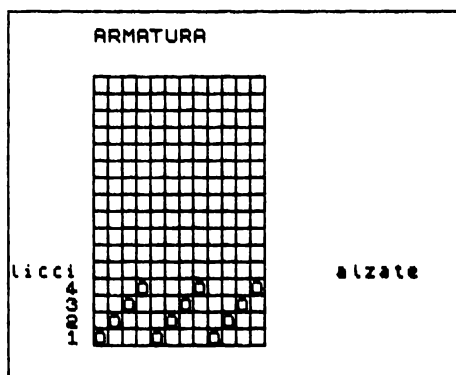
Isoliamo un campo formato da dodici righe e da dodici colonne con la convenzione che le righe rappresentano i fili di trama e le colonne i fili di ordito.

Inferiormente a quest'area di disegno del tessuto, isoliamo altre quattro linee quadrettate al cui lato, partendo dal basso, riportiamo il numero progressivo dei licci.

Per l'assegnazione di ogni singolo filo di ordito al rispettivo liccio, secondo una regola assegnata, scriviamo una "o" nel quadretto d'incrocio tra la colonna corrispondente al filo di ordito e la riga di quadretti relativa al liccio a cui appartiene la maglia.

Riportiamo la scrittura della alzata a fianco di tutte le righe della messa in carta comprese nel modulo.

In definitiva lo schema dovrà assumere la seguente forma:



Affrontiamo ora il problema della campitura dell'area dello schema destinata al disegno dell'intreccio.

La campitura, intendendo con essa l'operazione di annerimento di quadretti su un piano, è funzione del tipo di intreccio; per cui ad ogni armatura corrisponde una ed una sola messa in carta.

Per ciascuna armatura allora occorre accedere al relativo spazio di memoria nella matrice t\$, controllare il valore matematico delle cifre ivi memorizzate e in funzione di esso annerire un quadretto ogni volta che si incontra un 1.

Nella messa in carta però vogliamo fare apparire i due sistemi di codifica dell'intreccio che abbiamo scelto.

Limitiamo la scrittura delle cifre binarie alle righe di quadretti contenute nel modulo delle singole armature e destiniamo alle restanti righe di quadretti la codifica grafica.

Le due procedure ora descritte vengono eseguite ogni volta che si vuole visualizzare una messa in carta.

Con le istruzioni che le realizzano pertanto abbiamo creato due sottoprogrammi con lo scopo di:

- snellire la procedura di cui ci stiamo occupando;
- risparmiare memoria RAM.

Veniamo alla codifica BASIC di questi due sottoprogrammi.

Alle istruzioni del primo abbiamo riservato i numeri compresi tra 510 e 595. Il listato che lo realizza è il seguente.

```
500>REM Sottoprogrammi
510 LET d$="licci": LET z$="alz
a:
515 REM Schema per armature
520 FOR i=1 TO 136 STEP 3
525 PLOT 48,15+i: DRAW 96,0
530 NEXT i
540 FOR i=1 TO 104 STEP 3
550 PLOT 47+i,16: DRAW 0,128
555 NEXT i
560 FOR i=0 TO 11 STEP 4: FOR j
=0 TO 3
565 PRINT OVER 1:AT 19-j,6+j+i;
"O": NEXT j: NEXT i
580 PRINT AT 15,0;d$:AT 15,23;z
$
590 PRINT AT 16,4;4:AT 17,4;3;A
T 18,4;2:AT 19,4;1
595 RETURN
```

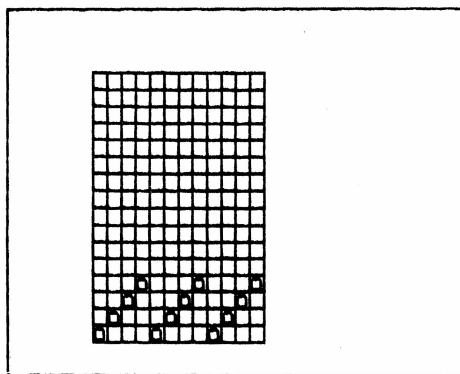
Le istruzioni da 520 a 555 disegnano le linee orizzontali e quelle verticali, incrociandosi, realizzano la quadrettatura.

Lo spazio compreso tra le linee verticali rappresenta i fili d'ordito, quello compreso tra le linee orizzontali i fili di trama.

Con la doppia iterazione, controllata dalle memorie i e j (linee di programma 560 e 565), scriviamo un "O" nei quadrettini delle ultime quattro righe di quadretti in accordo con quanto detto in precedenza.

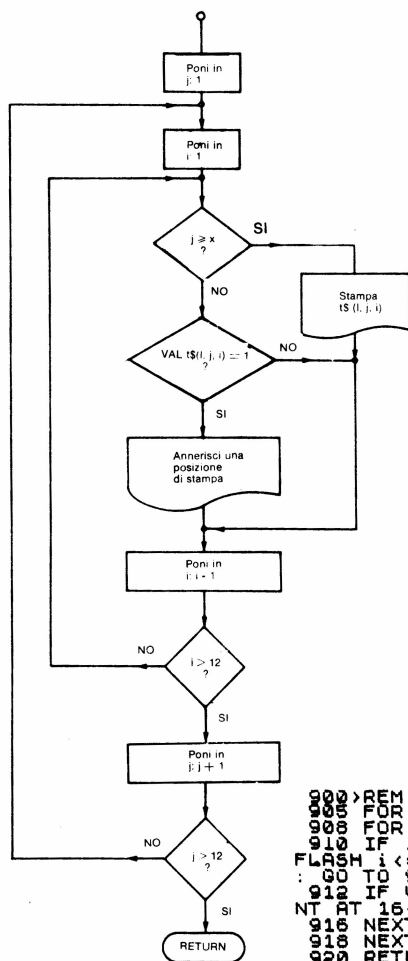
Il comando OVER 1 nell'istruzione di stampa fa sì che la scrittura delle "O" nel quadretto non cancelli i lati del quadratino stesso.

Il risultato che si ottiene è mostrato in figura:



Le istruzioni 580 e 590 completano lo schema (vedi figura 3.1.2.8).

Del secondo sottoprogramma, che per brevità chiameremo "campiture", diamo il diagramma di flusso e la codifica BASIC (linee di programma da 900 a 920):



```

9000>REM Campitura
9001  FOR j=1 TO 16
9002  FOR i=1 TO 16
9003  IF j<=x THEN PRINT OVER 1;
9004  FLASH i<=f; AT 16-j,5+i;t$(l,j,i)
9005  : GO TO 9016
9006  IF VAL t$(l,j,i)=1 THEN PRI
9007  NT AT 16-j,5+i;"■";
9008  NEXT i
9009  NEXT j
9010  RETURN
  
```

La condizione logica $j \leq x$ descritta nella linea di programma 910 controlla la scrittura in codice simbolico (0 e 1) delle x righe comprese nel modulo dell'armatura in esame.

Mentre il soddisfacimento della condizione $VAL\ t\$ (l,j,i) = 1$ descritta nella linea 912, consente di annerire un quadretto nell'area di disegno della messa in carta.

Le due condizioni consentono quindi la scrittura di tessuti secondo i due sistemi di codifica scelti simulando l'operazione di campitura di una superficie.

Infatti l'istruzione 912 consente la stampa di un quadratino nero tutte le volte che il valore matematico dell'elemento corrente delle matrici $t\$$ definito dai valori di l, j e i è 1.

Reiterando le istruzioni 910 e 912 per 144 volte (12×12) si visualizza nei due sistemi di codifica il tipo di tessuto fissato dal valore di 1.

Il comando FLASH presente nell'istruzione 910 controlla la stabilità delle f cifre sulla riga j -esima.

La verità della condizione $i \leq f$ fa sì che FLASH assuma il valore 1 determinando il lampeggiamento del modulo di riga, al contrario se risulterà falsa assegnerà a FLASH il valore 0 stabilizzando le restanti cifre della riga.

Questo procedimento ripetuto x volte e controllato dalla condizione $j \leq x$ produce il lampeggiamento del modulo nella messa in carta (non nascondiamo la nostra soddisfazione per il risultato ottenuto!)

Alle volte può essere comodo poter riprodurre su carta la videata corrispondente alle messe in carta.

Per questo motivo abbiamo creato un semplice sottoprogramma preposto a questo scopo.

Il listato è il seguente:

```
700>REM Per copiare
720 INPUT "Per COPIARE inserisc
i caratteri un qualunque cara
tere" a$
730 IF a$="c" OR a$="C" THEN CO
PY RETURN
740 RETURN
```

Anche per il sottomenù visualizzato dal programma principale della procedura "visualizzazione nella messa in carta" abbiamo creato un sottoprogramma dovendo essere ripetuto anche nella procedura "visualizzazione del tessuto colorato".

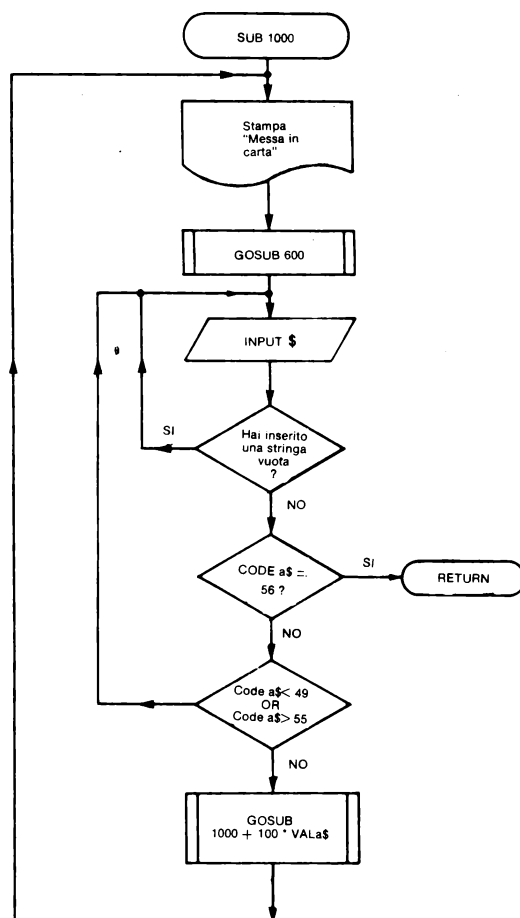
Le istruzioni che lo realizzano sono:

```
600>REM Sottomenu'
610 PRINT AT 3,5;"1 -per TELA
615 PRINT AT 5,5;"2 -per NATTE'
620 PRINT AT 7,5;"3 -per REP d'
ordito"
625 PRINT AT 9,5;"4 -per REP di
trama"
630 PRINT AT 11,5;"5 -per SAIA
635 PRINT AT 13,5;"6 -per BATAV
IA
640 PRINT AT 15,5;"7 -per RASO
645 PRINT AT 17,5;"8 -per USCIR
e
650 PRINT AT 20,1;"seleziona in
serendo un numero"
660 RETURN
```

A questo livello di raffinamento il problema è sufficientemente "frammentato".

Possiamo allora passare alla codifica BASIC delle procedure del sottoproblema "visualizzazione e stampa delle messe in carta".

Incominciamo con il proramma principale in cui diagramma di flusso è mostrato in figura:



La cifra 8 (codice 56) consente di uscire da questa procedura e di ritornare al menù principale.

L'immissione in a\$ di stringhe nulle e di caratteri con codice minore di 49 o maggiori di 55 non vengono accettati: si ha quindi accesso alla chiamata dei sottoprogrammi solo quando vengono inserite le cifre da 1 a 7.

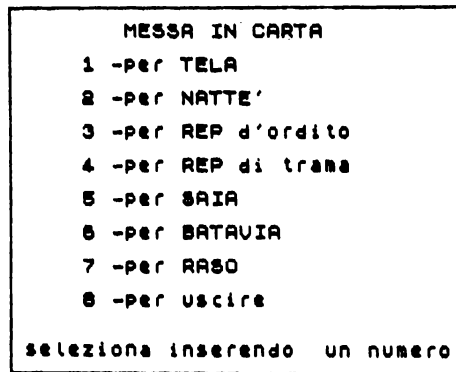
La regola di rinvio ai sottoprogrammi è $1000 + 100 * \text{"valore matematico di a$"}.$

La lista di istruzioni è elencata nel seguente listato.

```

1000>REM Messa in carta
1010 BRIGHT 1: PRINT FLASH 1;AT
1,8,"MESSA IN CARTA"
1020 GO SUB 600
1050 INPUT a$
1065 IF a$="" THEN GO TO 1050
1068 IF CODE a$=56 THEN RETURN
1070 IF CODE a$<49 OR CODE a$>55
THEN GO TO 1050
1080 GO SUB 1000+100*VAL a$
1090 CLS : BRIGHT 0: GO TO 1000
  
```

Il sottomenù visualizzato è:

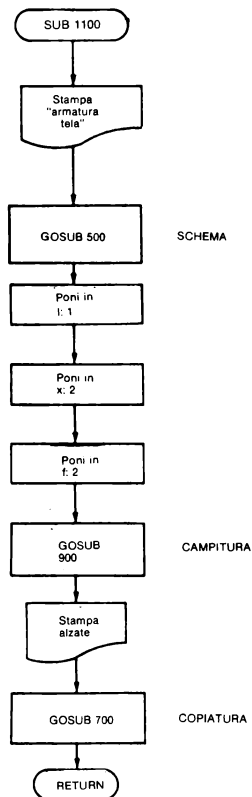


Al ritorno dai sottoprogrammi, selezionati inserendo una cifra compresa tra 1 e 7, l'esecuzione dell'istruzione GOTO 1000 contenuta nella linea 1080 consente il salto alla linea 1000 e quindi la nuova scrittura del sottomenù.

La cifra 8 come detto consente il ritorno al menù principale.

Messa in carta dell'armatura tela

Rappresentiamo mediante un diagramma di flusso la procedura risolutiva.



Vediamo la funzione delle memorie l, x e f.

Come detto in precedenza, per campire l'area di disegno nello schema per la messa in carta è necessario controllare il valore dei dati presenti nella matrice t\$ che, ricordiamolo, memorizza in forma codificata (cifre zero e uno) la posizione reciproca dei fili di trama e di ordito all'atto del loro incrocio.

Assegnando il valore 1 alla memoria l, l'elaborazione nel sottoprogramma 900 si baserà sull'elemento t\$ (1,18,24) della matrice t\$, quello cioè relativo alla tela.

Alla memoria x assegnamo il valore pari al numero di righe presenti nel modulo, cioè 2.

Possiamo così far stampare due righe in cifre binarie.

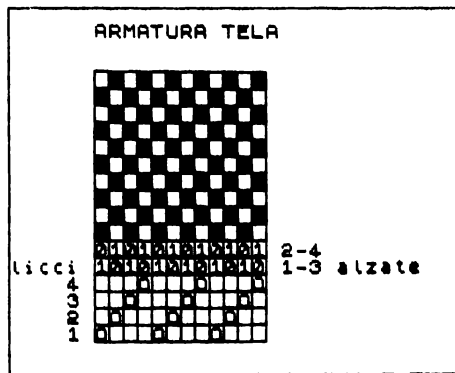
Il valore assegnato alla memoria f controlla il lampeggiamento del modulo di riga dell'armatura.

L'azione combinata dei valori di x e f consentirà il lampeggiamento del modulo di superficie.

Segue il listato:

```
1100>CLS : REM Tela
1105 PRINT AT 1,6;"ARMATURA TELA"
1106 GO SUB 500
1110 LET l=1: LET x=2: LET f=2
1120 GO SUB 900
1135 PRINT AT 15,19;"1-3";AT 14,
19;"2-4"
1150 GO SUB 700
1190 RETURN
```

e la videata corrispondente:



Messa in carta per l'armatura nattè

La struttura di questo sottoprogramma, e di tutti quelli che seguiranno, è identica a quella relativa all'armatura tela.

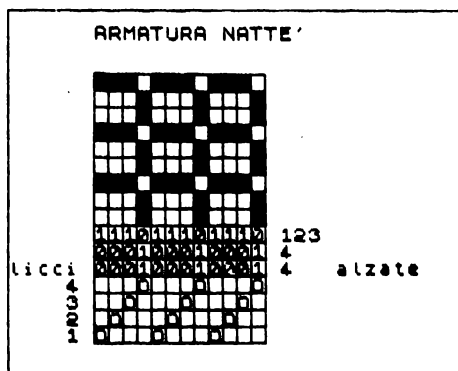
Quindi ci limitiamo a fornire il listato di ciascuna di esse e la videata corrisponde.

Si rifletta sui valori di l, x e f.

Listato:

```
1200>CLS : REM Nattè
1210 PRINT AT 1,6;"ARMATURA NAT"
1215 GO SUB 500
1220 LET l=2: LET x=3: LET f=4
1230 GO SUB 900
1250 PRINT AT 15,19;"4";AT 14,19
14;"AT 13,19;"123"
1280 GO SUB 700
1290 RETURN
```

Videata:

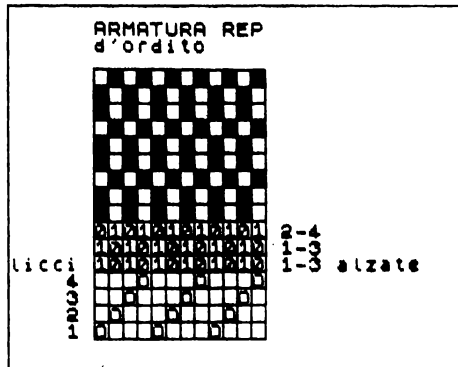


Messa in carta dell'armatura rep d'ordito

Listato:

```
1300>CLS : REM Rep d'ordito
1310 PRINT AT 1,6;"ARMATURA REP"
:AT 2,6;"d'ordito"
1312 GO SUB 500
1320 LET l=3: LET x=3: LET f=2
1330 GO SUB 900
1355 PRINT AT 15,19;"1-3";AT 14,
19;"1-3";AT 13,19;"2-4"
1360 GO SUB 700
1390 RETURN
```

Videata:

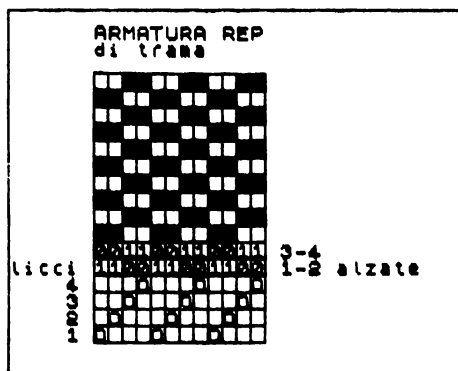


Messa in carta dell'armatura rep di trama

Listato:

```
1400>CLS : REM Rep di trama
1410 PRINT AT 1,6;"ARMATURA REP"
:AT 2,6;"di trama"
1412 GO SUB 500
1420 LET l=4: LET x=2: LET f=4
1430 GO SUB 900
1450 PRINT AT 15,19;"1-2";AT 14,
19;"3-4"
1460 GO SUB 700
1490 RETURN
```

Videata:

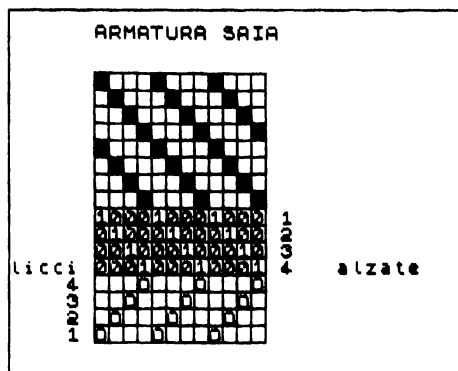


Messa in carta dell'armatura saia

Listato:

```
1500>CLS : REM Saia
1510 PRINT AT 1,6;"ARMATURA SAIA"
1515 GO SUB 500
1520 LET l=5: LET x=4: LET f=4
1530 GO SUB 900
1550 PRINT AT 15,19,4;AT 14,19;3
      AT 13,19;2;AT 12,19;1
1560 GO SUB 700
1590 RETURN
```

Videata:

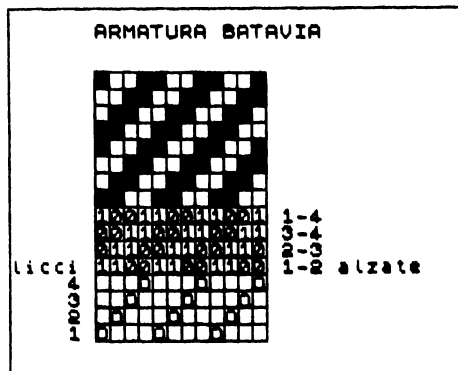


Messa in carta dell'armatura batavia

Listato:

```
1600>CLS : REM Batavia
1610 PRINT AT 1,6;"ARMATURA BATA
UIA "
1612 GO SUB 500
1620 LET l=6: LET x=4: LET f=4
1630 GO SUB 900
1655 PRINT AT 15,19;"1-2";AT 14,
19;"2-3";AT 13,19;"3-4";AT 12,19
;"1-4"
1660 GO SUB 700
1690 RETURN
```


Videata:

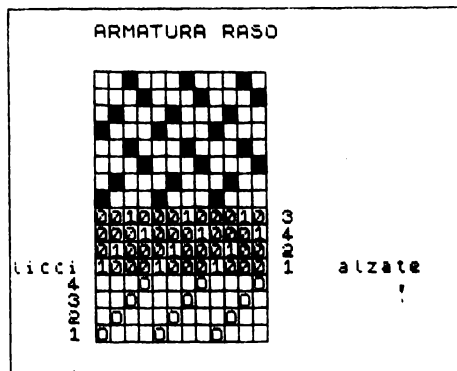


Messa in carta dell'armatura raso

Listato:

```
1700>CLS : REM Raso
1710 PRINT AT 1,6;"ARMATURA RASO"
1720 GO SUB 500
1730 LET l=7: LET x=4: LET f=4
1740 GO SUB 900
1770 PRINT AT 15,19;1;AT 14,19;2
;AT 13,19;4;AT 12,19;3
1780 GO SUB 700
1790 RETURN
```

Videata:



Per comodità del lettore riportiamo il listato completo di questo sottoprogramma.

```
1000>REM Messa in carta
1010 BRIGHT 1: PRINT FLASH 1;AT
1,8;"MESSA IN CARTA"
1020 GO SUB 600
1060 INPUT a$
1065 IF a$="" THEN GO TO 1060
1068 IF CODE a$=56 THEN RETURN
```

```

1070 IF CODE a$<49 OR CODE a$>55
    THEN GO TO 1060
1080 GO SUB 1000+100*VAL a$
1090 CLS : BRIGHT 0: GO TO 1000
1100 CLS : REM Tela
1105 PRINT AT 1,6;"ARMATURA TELA"
"
1106 GO SUB 500
1110 LET l=1: LET x=2: LET f=2
1120 GO SUB 900
1135 PRINT AT 15,19;"1-3";AT 14,
19,"2-4"
1180 GO SUB 700
1190 RETURN
1200 CLS : REM Nettezza
1210 PRINT AT 1,6;"ARMATURA NAT
TE"
"
1215 GO SUB 500
1220 LET l=2: LET x=3: LET f=4
1230 GO SUB 900
1260 PRINT AT 15,19;"4";AT 14,19
,"4";AT 13,19;"123"
1280 GO SUB 700
1290 RETURN
1300 CLS : REM Rep.d'ordito
1310 PRINT AT 1,6;"ARMATURA REP"
,"AT 2,6;"d'ordito"
1312 GO SUB 500
1320 LET l=3: LET x=3: LET f=2
1330 GO SUB 900
1355 PRINT AT 15,19;"1-3";AT 14,
19,"1-3";AT 13,19;"2-4"
1380 GO SUB 700
1390 RETURN
1400 CLS : REM Rep.di trama
1410 PRINT AT 1,6;"ARMATURA REP
","AT 2,6;"di trama"
1412 GO SUB 500
1420 LET l=4: LET x=2: LET f=4
1430 GO SUB 900
1450 PRINT AT 15,19;"1-2";AT 14,
19,"3-4"
1480 GO SUB 700
1490 RETURN
1500 CLS : REM Saia
1510 PRINT AT 1,6;"ARMATURA SAIA"
"
1515 GO SUB 500
1520 LET l=5: LET x=4: LET f=4
1530 GO SUB 900
1560 PRINT AT 15,19,4;AT 14,19,3
,AT 13,19,2;AT 12,19,1
1580 GO SUB 700
1590 RETURN
1600 CLS : REM Batavia
1610 PRINT AT 1,6;"ARMATURA BATA
VIA"
"
1612 GO SUB 500
1620 LET l=6: LET x=4: LET f=4
1630 GO SUB 900
1665 PRINT AT 15,19;"1-2";AT 14,
19,"2-3";AT 13,19;"3-4";AT 12,19
,"1-4"
1680 GO SUB 700
1690 RETURN
1700 CLS : REM Raso
1710 PRINT AT 1,6;"ARMATURA RASO"
"
1720 GO SUB 500
1730 LET l=7: LET x=4: LET f=4
1740 GO SUB 900
1770 PRINT AT 15,19,1;AT 14,19,2
,AT 13,19,4;AT 12,19,3
1780 GO SUB 700
1790 RETURN

```

SUB Routine 2000 visualizzazione a colori del tessuto

Le fasi fondamentali della risoluzione di questo problema sono le seguenti:

- 1) il disegno dell'ordito;
- 2) la simulazione del passaggio della trama nell'ordito.

Descriviamo l'algoritmo risolutivo di queste due procedure.

Disegno dell'ordito

A ciascuno dei 24 fili di ordito facciamo corrispondere una banda verticale nel colore assegnato al filo in esame.

Questo risultato è ottenibile variando gli attributi delle posizioni di stampa.

Nel nostro caso sarà sufficiente controllare il colore dell'inchiostro con il comando INK.

Usato come elemento di una istruzione di stampa, INK limita il suo effetto ai caratteri che vengono stampati con quella istruzione, senza influenzare le altre posizioni di stampa.

Il codice del colore da assegnare alla INK è letto dalla matrice c\$ nella quale precedentemente avevamo memorizzato il colore di tutti i fili di ordito e di trama.

In conclusione l'istruzione per la stampa dell'ordito assumerà la forma:

```
2370>PRINT INK VAL c$(1,J);AT 16  
-1,J-1;"■"
```

Volendo disegnare 24 bande verticali (quanti sono i fili di ordito) ognuna di 16 posizioni di stampa dobbiamo impostare il seguente programma:

```
2350>REM Disegno dell'ordito  
2360 FOR J=1 TO 24: FOR I=1 TO 1  
6  
2370 PRINT INK VAL c$(1,J);AT 16  
-1,J-1;"■"  
2380 NEXT I: NEXT J
```

Alle memorie di controllo della doppia iterazione, i e j, affidiamo il compito di "guidare" la stampa del quadratino sullo schermo, inizialmente posizionata nell'angolo basso a sinistra dell'area di disegno, dal basso verso l'alto e da sinistra a destra.

Simulazione del passaggio della trama nell'ordito

Il problema è analogo al precedente, sebbene in questo caso dobbiamo stampare dei quadretti orizzontali nel colore della trama, coerentemente con l'armatura di cui si vuole visualizzare il tessuto.

Il colore di ognuna delle 384 (24x16) posizioni di stampa dell'area destinata al disegno è funzione della posizione reciproca fra la trama e l'ordito all'atto del loro intreccio.

I dati relativi alle armature sono stati memorizzati nella matrice t\$; pertanto sarà il valore matematico di t\$ a stabilire se il colore delle singole posizioni dovrà essere cambiato oppure no.

Tutte le volte che tale valore è zero, ciò significa che l'ordito sta sotto la trama, allora si stamperà un quadratino del colore del filo di trama il cui codice è memorizzato in C\$.

Nell'effettuare tale controllo inviamo i messaggi allo schermo in modo tale che la posizione di stampa parta dall'angolo in basso a sinistra dell'area destinata alla visualizzazione del tessuto proseguendo verso destra e quindi verso l'alto, così come avviene nella scrittura dei tessuti.

Il risultato visivo che se ne ottiene, oltre ad essere altamente spettacolare, è assimilabile al percorso del filo di trama nell'ordito durante il funzionamento di un vero telaio.

Reiterando questa procedura per i sedici passaggi di trama previsti si ottiene una videata che mostra, nei colori scelti, la campionatura di una pezza.

Le istruzioni in basic che realizzano questo procedimento sono:

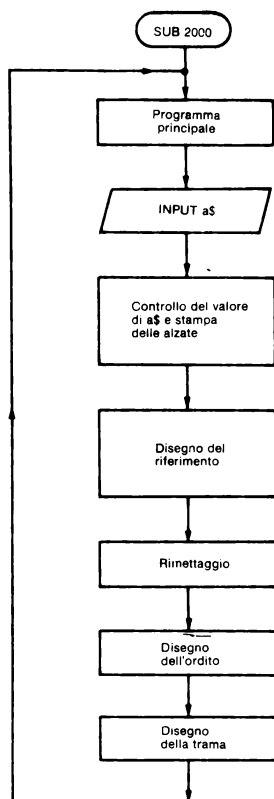
```
2385>REM Passaggio della trama n
ell'ordito
2390 FOR j=1 TO 16: FOR i=1 TO 2
4
2400 IF VAL t$(VAL a$(j,i))=0 THE
N PRINT INK VAL c$(2,j);AT 16-j,
i-1; "■"
2410 NEXT i: NEXT j
```

A questo punto il problema della simulazione dell'intreccio è risolto.

Non ci resta che controllare la variabile intreccio e organizzare un riferimento per la visualizzazione del tessuto con l'indicazione del rimettaggio e delle alzate.

Anche in questo caso per quanto detto è necessario un programma principale cui affidare il compito di visualizzare un sottomenù e il controllo del programma stesso.

La sua struttura assume pertanto la seguente forma:



Il programma principale controlla il programma e visualizza il seguente sottomenù:

TESSUTO COLORATO	
1	-per TELA
2	-per NATTE'
3	-per REP d'ordito
4	-per REP di trama
5	-per SAIA
6	-per BATAVIA
7	-per RASO
8	-per uscire
seleziona inserendo un numero	

Le istruzioni sono riportate sul listato:

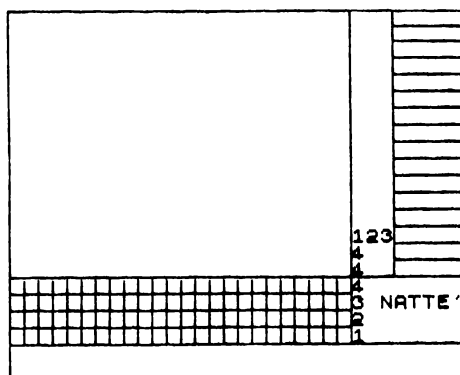
```
2000>: CLS : REM Tessuto colorat
0
2010 BRIGHT 1: PRINT FLASH 1;AT
1,7;"TESSUTO COLORATO"
2040 GO SUB 600
2050 INPUT a$
2060 IF a$="" THEN GO TO 2050
2065 IF CODE a$=56 THEN RETURN
2070 IF CODE a$<49 OR CODE a$>56
THEN GO TO 2050
2075 CLS
```

La procedura "controllo del valore di a\$ e stampa delle alzate" è codificata con la seguente listata di istruzioni.

```
2100>REM Scrittura delle alzate
2105 IF VAL a$=1 THEN PRINT AT 1
5,24;"1 3";AT 14,24;"2 4";AT 17,
206, FLASH 1;"TELA"
2110 IF VAL a$=2 THEN PRINT AT 1
5,24;4;AT 14,24;4;AT 13,24;123;A
T 17,26; FLASH 1;"NATTE"
2120 IF VAL a$=3 THEN PRINT AT 1
5,24;"1 3";AT 14,24;"1 3";AT 13,
24;"2 4";AT 17,26; FLASH 1;"REP"
2130 IF VAL a$=4 THEN PRINT AT 1
5,24;"1 2";AT 14,24;"3 4";AT 17,
206, FLASH 1;"REP tr"
2140 IF VAL a$=5 THEN PRINT AT 1
5,24;4;AT 14,24;3;AT 13,24;2;AT
12,24;1;AT 17,26; FLASH 1;"SAIA"
2150 IF VAL a$=6 THEN PRINT AT 1
5,24;"1-2";AT 14,24;"2-3";AT 13,
24;"3-4";AT 12,24;"1-4";AT 17,25
FLASH 1;"BATAVIA"
2160 IF VAL a$=7 THEN PRINT AT 1
5,24;1;AT 14,24;2;AT 13,24;4;AT
12,24;3;AT 17,26; FLASH 1;"RASO"
2165 PRINT AT 19,24;1;AT 18,24;2
;AT 17,24;3;AT 16,24;4
```

Il valore numerico di a\$ consente la stampa delle alzate cioè l'indicazione di quali sono i licci da azionare, in accordo con l'armatura scelta per tessere il tessuto che si vuole visualizzare.

Lo schema per il disegno del tessuto è mostrato in figura dove sono indicate esemplificativamente le alzate per il nattè.



Consente di scrivere:

- il rimettaggio nelle 4 righe quadrettate in basso;
- le alzate di fianco ai passaggi di trama sulla destra;
- il colore della trama nelle righe sulla destra;
- il tipo di armatura e il numero progressivo dei licci nel rettangolo in basso a destra.

La superficie bianca è destinata ad accogliere il disegno del tessuto.

Sia il rimettaggio che la trama sono stampati nel colore del rispettivo filo.

Il rimettaggio è ottenuto con le istruzioni da 2235 a 2260.

Esse producono la stampa di una o nel colore del filo di ordito, secondo un assegnato rimettaggio, nelle quattro righe quadrettate dello schema.

Le procedure-disegno dell'ordito e della trama sono già state spiegate.

Il listato completo del sottoprogramma 2000 è riportato nella figura seguente.

```

2000>: CLS : REM Tessuto colorat
2010 BRIGHT 1: PRINT FLASH 1;AT
1,7;"TESSUTO COLORATO"
2040 GO SUB 600
2050 INPUT A$
2060 IF A$="" THEN GO TO 2050
2065 IF CODE A$=55 THEN RETURN
2070 IF CODE A$<49 OR CODE A$>56
THEN GO TO 2050
2075 CLS
2100 REM Scrittura delle alzate
105 IF VAL A$=1 THEN PRINT AT 1
24,"1 3";AT 14,24;"2 4";AT 17,
6; FLASH 1;"TELA"
110 IF VAL A$=2 THEN PRINT AT 1
24,4;AT 14,24;AT 13,24;123;A
17,26; FLASH 1;"NATTE"
120 IF VAL A$=3 THEN PRINT AT 1
24,"1 3";AT 14,24;"1 3";AT 13,
4;"2 4";AT 17,26; FLASH 1;"REP"
130 IF VAL A$=4 THEN PRINT AT 1
24,"1 3";AT 14,24;"3 4";AT 17,
6; FLASH 1;"REP 1"
140 IF VAL A$=5 THEN PRINT AT 1
24,4;AT 14,24;AT 13,24;2;AT
24,1;AT 17,26; FLASH 1;"SAIA"
150 IF VAL A$=6 THEN PRINT AT 1
24,"1 2";AT 14,24;"2 3";AT 13,
4;"3 4";AT 12,24;"1 4";AT 17,26
FLASH 1;"BATAVIA"
160 IF VAL A$=7 THEN PRINT AT 1
24,4;AT 14,24;AT 13,24;4;AT
24,1;AT 17,26; FLASH 1;"RASO"
165 PRINT AT 19,24;1;AT 18,24;2
AT 17,24;3;AT 16,24;4
170 REM Schema di riferimento p
er tessuto
172 REM Linee orizzontali
175 FOR I=16 TO 175 STEP 8
178 IF I=16 THEN PLOT 0,16: DRA
W 255,0
2180 IF I<=48 THEN PLOT 0,I: DRA
W 192,0
2185 IF I>48 THEN PLOT 216,I: DR
AW 38,0
2188 IF I=64 THEN PLOT 0,48: DRA
W 255,0
2190 NEXT I
2192 REM Linee verticali
2195 FOR I=0 TO 56*4 STEP 8
2198 IF I<192 THEN PLOT I,16: DR
AW 0,30
2200 IF I=192 THEN PLOT 192,16:
DRAW 0,158
2205 IF I=216 THEN PLOT I,48: DR
AW 0,126
2210 NEXT I
2235 REM Rimettaggio
2240 FOR I=0 TO 23 STEP 4: FOR J
=0 TO 3
2250 PRINT OVER 1; INK VAL C$(1,
I+1);AT 19-J,J+I;"0"
2260 NEXT J: NEXT I
2270 REM trama
2280 FOR I=0 TO 15
2285 PRINT OVER 1; INK VAL C$(2,
J+1);AT 15-I,28;" "
2290 NEXT I
2300 REM Disegno dell'ordito
2350 FOR J=1 TO 24: FOR I=1 TO 1
2360 PRINT INK VAL C$(1,J);AT 16
-I-1;" "
2380 NEXT I: NEXT J
2385 REM Passaggio della trama n
ell'ordito
2390 FOR J=1 TO 16: FOR I=1 TO 2
2400 IF VAL t$(VAL A$,J,I)=0 THE
N PRINT INK VAL C$(2,J);AT 16-J,
I-1;" "
2410 NEXT I: NEXT J
2420 GO SUB 700
2450 CLS : BRIGHT 0: GO TO 2000

```

SUB Routine 4000 per uscire dal programma

Questo sottoprogramma, banale per la sua semplicità, ha lo scopo di dare all'utente la possibilità di abbandonare il programma in maniera "elegante".

Il listato è il seguente:

```
4000>REM Abbandono del programma
4005 CLS
4010 PRINT BRIGHT 1;AT 9,9;"TESS
ITURA"
4020 BRIGHT 0
```

3.2 Istogrammi: una utility per la geografia e le scienze

PIERO D'ALFONSO E GIOVANNI MARTORANA

3.2.1 La rappresentazione grafica

Esaminare dati per dedurre considerazioni è una abitudine che si sta radicando nella cultura di base tanto anche nella trattazione di argomenti scolastici quali si trovano nella geografia, nelle scienze e nell'educazione tecnica si procede consultando tabelle di dati quando queste tabelle non vengono addirittura costruite a partire dai valori rilevati nel corso di esperienze ed osservazioni.

Se da un lato l'abitudine a operare su dati garantisce una maggiore scientificità dello studio dall'altro offre però allo studente un quadro di lavoro molto arido: districarsi tra numeri, ordini di grandezze, valori percentuali non è sempre agevole e spesso si assiste ad errori sia di calcolo sia di lettura delle tabelle.

Questa è una delle ragioni per le quali sono state ampiamente utilizzate le tecniche di rappresentazione grafica che traducono visivamente le informazioni contenute nei dati.

La scelta della rappresentazione grafica dipende dalla nostra volontà:

- di evidenziare un evento,
- di fare la comparazione tra situazioni contemporanee,
- di studiare l'evolversi di un fenomeno,

a seconda dei casi si dimostrano più adatte alcune rappresentazioni piuttosto che altre, resta comunque vero che la possibilità di tradurre graficamente un insieme di dati dipende strettamente dai dati stessi e dalle elaborazioni che questi consentono.

Col presente lavoro si intende esaminare la capacità rappresentativa degli istogrammi, cioè di quei grafici che traducono i dati in colonne la cui altezza è proporzionale al loro valore.

In uno stesso sistema di riferimento è consentito inserire più istogrammi, con il risultato di permettere un confronto visivo immediato.

In proposito si hanno due possibilità a seconda che gli istogrammi siano stati costruiti:

- con grandezze uguali
- con grandezze disomogenee, ma appartenenti a una stessa classe.

Nel primo caso gli istogrammi mostrano l'evoluzione della grandezza rispetto ad un parametro comune quale potrebbe essere ad esempio l'andamento temporale, nel secondo caso invece gli istogrammi evidenzieranno le differenze esistenti tra le grandezze e forniranno gli elementi quantitativi necessari ad una comparazione; un esempio può essere in agricoltura il confronto tra le varie produzioni ospitate da una area definita quale la provincia oppure la regione o la nazione.

Si è già detto che l'altezza di un istogramma è proporzionale al valore della grandezza che rappresenta. Comunque la determinazione di questa proporzionalità merita alcune considerazioni.

Anche in questo caso si hanno due possibilità: o l'istogramma rappresenta direttamente il valore assoluto, oppure rappresenta il valore della grandezza espressa in percentuale.

Sull'asse verticale dovrà allora essere riportata o la scala dei valori assoluti, adeguata alle grandezze in esame con relativa unità di misura, oppure la scala dei valori percentuali.

Almeno due fattori giocano a favore della rappresentazione ottenuta con valori percentuali:

- la scala di riferimento dei valori assoluti necessita continue modifiche in dipendenza delle variazioni degli ordini di grandezza e delle unità di misura dei dati esaminati, invece la scala di riferimento dei valori percentuali è sempre la stessa (da 0% a 100%) e non dipende dall'unità di misura;

- qualora si dovessero confrontare istogrammi inseriti in differenti sistemi di riferimento se i riferimenti sono assoluti, prima di esaminare le diverse altezze degli istogrammi si deve procedere ad accordare le scale di riferimento rispetto ai valori massimi. Ciò non accade se i riferimenti sono percentuali in quanto le scale di riferimento sono le stesse.

La facilità di lettura degli istogrammi espressi in valori percentuali non deve però trarre in inganno infatti affinché i valori rappresentati assumano un significato oggettivo, bisogna che sia evidenziato il valore globale cui quelle percentuali si riferiscono, completo di unità di misura: solo in questo modo sarà significativo operare confronti tra istogrammi di uno stesso sistema di riferimento ovvero trasferire istogrammi su altri sistemi di riferimento.

Infine un'ultima considerazione riguarda l'accettabilità di valori percentuali degli stessi istogrammi contenuti in uno stesso sistema di riferimento.

L'operazione di accostare più istogrammi induce al confronto; pertanto è abituale mettere insieme più istogrammi relativi ad un comune valore complessivo.

Risultato vuole che la somma dei valori percentuali rappresentati non superi il 100%, percentuale che come noto corrisponde proprio al valore complessivo.

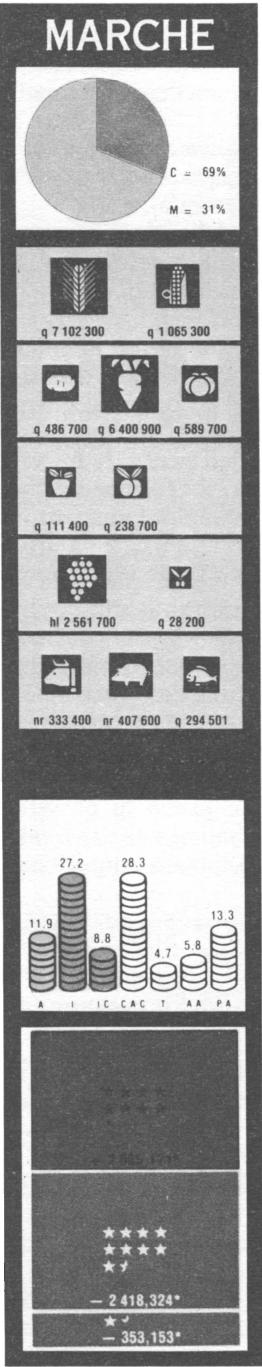
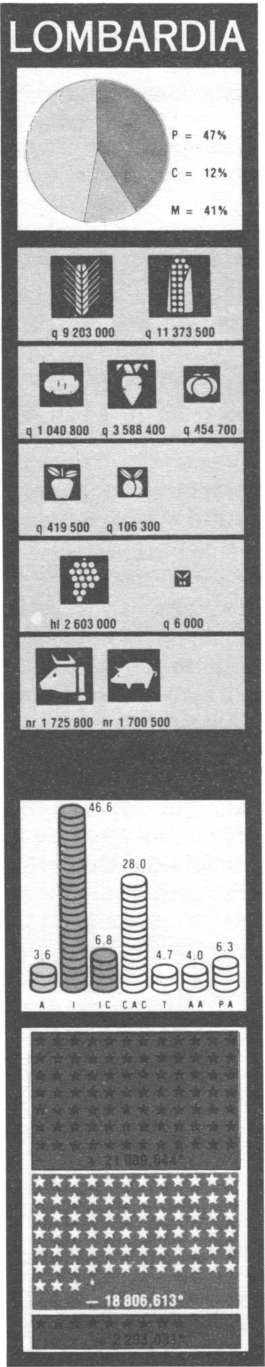
D'altra parte è pur vero che non tutti i dati che contribuiscono a totalizzare il valore complessivo devono essere necessariamente rappresentati con il relativo istogramma, quindi potrebbe benissimo verificarsi che la somma dei valori percentuali rappresentati con istogrammi risulti inferiore al 100%.

Nel paragrafo successivo affrontiamo un caso concreto in cui tutte queste considerazioni trovano un riscontro diretto.

3.2.2. Le statistiche socio economiche nell'atlante di lavoro di Giulio Mezzetti

L'Atlante di lavoro è una pubblicazione che fa parte integrante del corso di Geografia scritto da Giulio Mezzetti per la scuola media, in questo paragrafo prendiamo in considerazione le tabelle dei confronti regionali d'Italia sotto riprodotto.

Le tabelle, spiega l'autore, sono frutto di una sua originale elaborazione dei dati ottenuti da fonti ufficiali quali il



Estratto dalle pagine 28-29 dell'Atlante di lavoro del Testo "Geografia" di Giulio Mezzetti, La Nuova Italia Editrice, Firenze 1979.

- IX° censimento generale della popolazione (1971);
- i Conti Economici Generali delle Camere di Commercio (1976);
- compendio Statistico Italiano (1976).

Nelle tabelle gli fa ampio uso della comunicazione visiva realizzata con rappresentazioni ad

- areogrammi;
- istogrammi;
- ideogrammi.

Ognuna delle quali comprende l'indicazione quantitativa che la riguarda:

- valori percentuali,
- valori assoluti.

Le tabelle sono strutturate a doppia entrata: in testa sono riportati i nomi delle regioni italiane, sul lato sono collocati i dati; all'incrocio è posta la rappresentazione grafica.

Oltre alla lettura diretta di tale rappresentazione le tavole contengono tutti i dati necessari ad operare confronti

- tra regioni per uno stesso dato,
- tra dati per varie regioni;



ed inoltre a tradurre i valori percentuali in valori assoluti e viceversa.

Un esempio è suggerito dall'autore stesso a riguardo della FORMAZIONE DEL REDDITO specificatamente per il settore agricolo per il quale intende operare un confronto tra le produzioni della Lombardia e delle Marche.

Un confronto diretto tra gli istogrammi per l'AGRICOLTURA attribuirebbe un 11,9% alle Marche contro un 3,6% della Lombardia, in altre parole risulterebbe che le Marche producono un maggiore reddito sui prodotti agricoli di quanto non produca la Lombardia: l'affermazione è quanto meno assurda date le caratteristiche morfologiche e le differenze di superficie nettamente a favore della Lombardia.

Di fatto nel paragrafo precedente ci si è a ragione preoccupati di sottolineare che la facilità di lettura degli istogrammi espressi con valori percentuali non deve trarre in inganno: prima di passare a qualunque considerazione è necessario riferire i valori percentuali ai rispettivi valori globali.

Ciò è quanto l'autore propone nella nota che riproduciamo sotto:

<p>q produzione 1974 in quintali</p> <p>hi produzione 1974 in ettolitri</p> <p>nr n° capi allevati 1974</p> <p>produzione 1974 in miliardi di lire correnti. Es: 22.459 va- letto 22 miliardi e 459 milioni</p>	<p> la grandezza dei simboli è proporzionale alle quantità espresse in cifre</p> <p>★ 250 miliardi di lire prodotti a prezzi correnti 1974</p> <p>★ 250 miliardi di lire impiegati a prezzi correnti 1974</p>	<p> Ogni moneta vale L. 20. Le pile di monete indicano il contributo dato dai vari settori produttivi alla formazione di ogni L. 1.000 di reddito. Per esempio, su L. 1.000 di reddito prodotte in Lombardia, L. 466 (pari a circa 23 monete del grafico) sono dovute all'industria, e solo L. 36 all'agricoltura. Invece su L. 1.000 prodotte nelle Marche, L. 272 (pari a circa 14 monete) sono dovute all'industria e L. 119 all'agricoltura. Il reddito complessivo della Lombardia è però di L. 21.098 miliardi, quello delle Marche è di L. 2.065 miliardi (bilancio regionale, ultima casella). Quindi l'agricoltura lombarda produce complessivamente L. 21.098 miliardi x 36% = 759 miliardi mentre l'agricoltura marchigiana produce L. 2.065 miliardi x 119% = 246 miliardi</p>
--	--	---

Si capisce quindi quanto sia importante disporre in questo esempio del reddito lordo di ciascuna regione, senza questo dato il confronto tra regioni sarebbe stato impossibile; tanto è vero che per non averne volutamente tenuto conto si era arrivati ad una conclusione errata.

L'esempio appena trattato fa capire quanto possa risultare complesso operare sui dati quando questi non sono già stati elaborati, d'altra parte è proprio compito delle discipline scolastiche insegnare a produrre elaborazioni al fine di fornire i necessari supporti alle tesi che si vogliono dimostrare.

Fare una elaborazione di dati comporta da un lato la capacità progettuale di costruire una informazione significativa da un complesso di dati, dall'altro provvedere operativamente con

- procedure di calcolo,
- rappresentazioni grafiche;

al fine di rendere l'informazione disponibile e comunicabile.

Mentre il soggetto della progettazione è e deve restare il docente, che curerà personalmente di metterne a conoscenza gli studenti, tutte le procedure operative devono poter essere svolte autonomamente dagli studenti stessi. È a queste procedure che l'uso dell'elaboratore può essere di qualche utilità, ed è nell'esercizio d'uso dei programmi che gli studenti devono cimentarsi.

Il programma che viene presentato nei paragrafi successivi è un esempio di "programma di UTILITÀ", per costruire istogrammi a partire da una tabella di dati.

3.2.3. La tabella di comparazione e gli istogrammi

Per descrivere il criterio di costruzione di una tabella in primo luogo va precisato in che modo vengono identificati i dati: abitualmente bastano due elementi e cioè

- il nome del dato,
- la variabile.

La variabile consente di assegnare al dato un valore infatti il rilevamento del dato dipende proprio dalla variabile scelta; per fare un esempio basta riconsiderare la tavole dell'Atlante:

- il reddito agricolo è il nome di un dato che può essere quantificato solo dopo che è stata assegnata la *variabile*: regione d'Italia.

Un altro elemento di identificazione del dato è l'unità di misura in cui sono espressi i valori, è però buona norma considerare l'unità di misura parte integrante del dato stesso.

In conclusione ci troviamo ad avere dati predisposti ad essere collocati all'interno di tabelle a doppia entrata le cui griglie sono determinate dai *nomi dei dati* e dalle *variabili*.

Una tabella potrà essere utilizzata per più dati a condizione che i dati siano riferiti ad un comune insieme di variabili.

A differenza di quanto detto a favore degli istogrammi in cui i valori sono espressi in percentuale, nella tabella dovranno essere ospitati solo valori assoluti in quanto è la disponibilità di questi valori che potrà consentire in seguito di ricavarne i valori percentuali con i quali costruire gli istogrammi.

Il prelievo dei dati dalla tabella può avvenire in più modi e precisamente:

- si ottengono i valori di uno stesso dato per differenti variabili qualora i dati vengano prelevati dalle caselle poste sulla stessa linea orizzontale,
- si ottengono i valori di più dati per differenti variabili qualora i dati vengano prelevati da caselle poste lungo le linee orizzontali che hanno per intestazione il nome dei dati richiesti in corrispondenza delle linee verticali intestate con le variabili scelte.
- si ottengono i valori di più dati per una stessa variabile qualora i dati vengano prelevati dalle caselle poste su una stessa linea verticale.

Per ogni modalità di prelievamento dei dati possono essere costruiti gli istogrammi non appena viene determinato il valore globale su cui vanno calcolati i valori percentuali.

Il valore globale può essere:

- la somma di tutti i valori assoluti dei dati con cui si intendono costruire gli istogrammi,
- la somma dei valori assoluti di un insieme di dati nel quale sono tra gli altri compresi quelli dei dati da rappresentare con istogrammi.

Uso possibile del programma istogrammi

Come ogni UTILITY il programma che qui si presenta ha una possibilità di utilizzo ampia e non limitata al singolo caso.

Non ha il compito di fare gli istogrammi di un certo libro (in questo caso quello di Giulio Mezzetti) o di un certo caso, ma di organizzare una "base di dati" (database) in modo tale da rendere possibile l'organizzazione di alcune correlazioni e la loro rappresentazione grafica in istogrammi.

L'uso quindi di quel testo specifico, lo si capirà procedendo nella lettura, è esemplificativo.

Si può forse dire, a suo merito, che è il testo in cui è stato più facile per noi studiare, estrarre e correlare dati.

Occorre dunque organizzare una gestione del programma secondo questo schema (o uno schema analogo):

1. Carica il programma generale dal cartridge al computer.
2. Togli il cartridge del programma generale e sostituisilo con un cartridge appena formattato che abbia una sigla che tiene conto del contenuto.
3. Carica i dati sul cartridge attraverso le funzioni che trovi nel Menù del programma. Puoi spegnere il computer.
4. Fai gli istogrammi attraverso le funzioni che trovi nel menù del diagramma. Se il computer era stato spento:
 - esegui l'operazione 1,
 - togli il cartridge del programma generale e sostituisilo con quello la cui sigla risponde alle tue esigenze,
 - puoi eseguire l'operazione 3,
 - ora sei all'operazione 4.

Puoi spegnere il computer.

Di seguito descriviamo quanto accadrebbe in un'aula in cui gli studenti e il loro professore, a conoscenza delle modalità d'uso del programma, lo interrogassero e/o lo informassero di dati riguardanti la geografia di un ben preciso territorio.

“Conoscenza delle modalità d'uso” è da intendere come conoscenza dei menu e delle relative correlazioni; a tale risultato si può arrivare sia consultando eventuali manuali d'uso, che è la strada comunemente offerta dai costruttori di software, sia venendo direttamente a conoscenza delle modalità risolutive del problema impiegate dal softwareista responsabile della scrittura del programma: questa è la strada che abbiamo percorso noi e la riportiamo integralmente nel paragrafo successivo.

Questa scelta va nella direzione di offrire all'utente i mezzi per introdurre modifiche migliorative non solo nella gestione del programma, ma nel programma stesso.

Premettiamo alcune indicazioni operative per consentire una precoce familiarizzazione con i menu.

Dopo aver caricato il programma da cartridge, sul video compare:

```

      ISTOGRAMMI
PER RICHIAMARE DATI PREMI -> 1
PER INSERIRE DATI PREMI ---> 2
PER DISEGNARE ISTO PREMI --> 3
PER LISTARE DATI PREMI ----> 4
PER CONSERVARE DATI PREMI -> 5
PER CORREGGERE PREMI -----> 6
USCIRE DAL PROGRAMMA PREMI > 7

SCEGLI UN NUMERO POI PREMI ENTER
```

In memoria non è contenuto alcun dato, quindi ci sono due strade da seguire:

- o si richiamano i dati,
- o si inseriscono dati.

Nel caso in cui si voglia richiamare dati si è posti di fronte alle alternative:

```

      RICHIAMARE DATI

per richiamare la tabella
dell'istogramma premi -----> 1

per richiamare la tabella
GENERALE premi -----> 2

per impostare la tabella
GENERALE premi -----> 3

per tornare al MENU premi -> 0

SCEGLI UN NUMERO POI PREMI ENTER

```

Le prime (1 e 2) fanno riferimento a dati conservati in precedenza sul cartridge dei dati (suggeriamo di avere due cartridge: uno per i dati e uno per il programma), la terza consente di predisporre una nuova Tabella GENERALE pronta a ricevere dati.

Se invece si sceglie la strada di inserire dati, le alternative sono:

```

      INSERIRE DATI

per inserire dati direttamen
te in tabella di istogramma
premi -----> 1

per inserire dati in
tabella GENERALE premi ----> 3

per inserire dati in tabella
di istogramma prelevandoli
da tabella GENERALE premi -> 5

per tornare al MENU premi -> 0

SCEGLI UN NUMERO POI PREMI ENTER

```

In proposito va detto che solo la 1 è consentita senza dati già in memoria, infatti:

- per la 3 si danno due casi:
 - o si è caricata da cartridge la Tabella GENERALE,
 - o si è appena impostata una nuova Tabella GENERALE come anticipato nel precedente menu,
- per la 5 è possibile operare solo a patto di disporre in memoria di una Tabella GENERALE già contenente dati, infatti questa opzione opera un trasferimento dati dalla Tabella GENERALE alla tabella degli istogrammi che è possibile solo nelle condizioni dette.

3.2.4. Elaborazioni

Per esemplificare l'impiego di questo programma di utilità, che ha la capacità di trattare dati numerici e tradurli in istogrammi, prefiguriamo come condizioni di lavoro quelle del docente di geografia che con la propria classe studia territorio, produzione, socialità ed economia dell'Italia settentrionale e dispone dell'Atlante di Lavoro più volte citato.

Man mano che procede con l'esame delle varie componenti geografiche rileva dalle tabelle socio-economiche i dati che gli servono a quantificare gli elementi trattati e utilizzando l'opzione INSERIRE DATI in Tabella GENERALE provvede a riempire la tabella con le informazioni corrispondenti.

La Tabella GENERALE è strutturata per VARIABILI e DATI, nel caso specifico alla voce VARIABILI saranno associati i nomi delle regioni, alla voce DATI gli elementi territoriali, sociali, produttivi, economici; con la tecnica di inserimento in tabella a doppia entrata vengono pertanto collocati i valori dei DATI per ogni VARIABILE (REGIONE) interessata.

L'accesso alla tabella è diretto e non sequenziale (in termine tecnico random), quindi i valori possono essere inseriti in momenti diversi senza problemi e senza creare conseguenze ai valori già inseriti; in altre parole il docente può seguire sia la strada che lo porta a trattare regione dopo regione, sia quella che lo conduce attraverso le regioni nominandone una dopo l'altra le varie componenti geografiche; in entrambi i casi non appena incontra DATI quantificabili è in grado di comunicarli all'elaboratore.

Le tabelle dell'Atlante di lavoro sono a questo fine un prezioso magazzino di valori dal quale è agevole il prelievo; in proposito va ricordato che il programma di utilità è progettato per ricevere sia valori assoluti sia valori percentuali.

Ammettiamo quindi che la Tabella GENERALE sia stata man mano riempita di valori in corrispondenza delle VARIABILI (REGIONI) e dei DATI (elementi geografici) trattati, esamineremo di seguito alcune elaborazioni che il programma di utilità consente agli studenti di eseguire.

Confronto diretto tra regioni

Il territorio esaminato sia la pianura padana e l'oggetto del confronto sia la produzione di cereali: si vuole conoscere l'incidenza di tale produzione nelle regioni

Piemonte,
Lombardia,
Emilia-Romagna,
Veneto.

L'analisi procederà dal prelievo di DATI della Tabella GENERALE le produzioni da considerare sono quelle del grano e del mais; operativamente si dovrà scegliere nel Menù principale l'opzione:

PER INSERIRE DATI

e nel Menù secondario

```

                INSERIRE DATI

per inserire dati direttamen
te in tabella di istogramma
premi -----> 1

per inserire dati in
tabella GENERALE premi ----> 3

per inserire dati in tabella
di istogramma prelevandoli
da tabella GENERALE premi -> 5

per tornare al MENU premi -> 0

SCEGLI UN NUMERO POI PREMI ENTER

```

l'opzione contraddistinta dal numero 5, dopo di che il computer ci richiederà tutte le informazioni che gli necessitano per costruire la tabella dell'istogramma. Seguiamo il dialogo tra computer ed utente:

```

Spectrum : scrivi il titolo dell'istogramma " L "
utente   : grn/pd
Spectrum : quante variabili ti occorrono?
           (massimo consentito 7)
utente   : 4
Spectrum : scrivi il nome delle variabili " L "
           (n.d.a.: per ogni variabile va premuto ENTER)
utente   : piemonte (ENTER)
           lombardia (ENTER)
           emilia-romagna (ENTER)
           veneto (ENTER)
Spectrum : quanti dati inserisci per ogni variabile?
           (massimo consentito 1)
utente   : grano

```

Giunti a questo punto sul video sono man mano comparse le seguenti scritture che confermano all'utente la ricezione delle informazioni da lui inviate allo SPECTRUM:

```

      grn/pd
4          VARIABILI
piemonte
lombardia
emilia-romagna
veneto
1          DATI
grano

```

Un breve commento spetta all'assegnazione del titolo per l'istogramma: nonostante potesse essere scritto per esteso (grano/padana) lo si è comunicato in una forma contratta a sei caratteri privandolo delle vocali; in questo modo il nome è predisposto ad essere archiviato nel cartridge, che consente nomi di registrazioni di dieci caratteri al massimo, anticipando che l'archiviazione della tabella dell'istogramma prevede le terminazioni standard dt1, dt2, dt3, dt4 si capisce la ragione della scelta proposta.

Torniamo allo Spectrum ed all'utente.

Sul video sono comparsi i valori che il computer ha prelevato dalla tabella Generale, poiché la scrittura non era tutta contenibile sulla stessa pagina questa operazione viene effettuata con uno scroll verticale, cioè uno scorrimento verso l'alto della pagina video.

```

emilia-romagna
veneto
1          DATI
grano
1) grano q    piemonte
              8485888
2) grano q    lombardi
              9283888
3) grano q    emilia-r
              13488788
4) grano q    veneto
              7878988
PREMI UN TASTO PER PROSEGUIRE

```

L'utente in base alla richiesta preme un tasto e lo Spectrum gli propone di scegliere tra le seguenti opzioni quella che ritiene più adeguata a fissare il valore globale su cui verranno poi determinati i valori percentuali da evidenziare nell'istogramma.

```

VALORE GLOBALE DATI

.
se valore globale e' somma
dei dati inseriti premi ---> 1

se inserisci valore globale
direttamente premi -----> 2

se prelevi valore globale da
tabella GENERALE premi ----> 3

SCEGLI UN NUMERO POI PREMI ENTER

```

Nel nostro caso l'utente sceglie: 1.

Spectrum: valore globale dei dati:

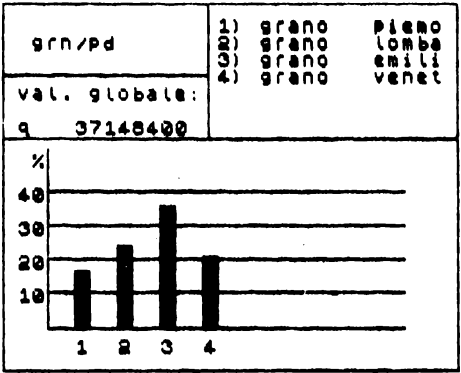
q 37148400

ancora?(s/n)

utente: n

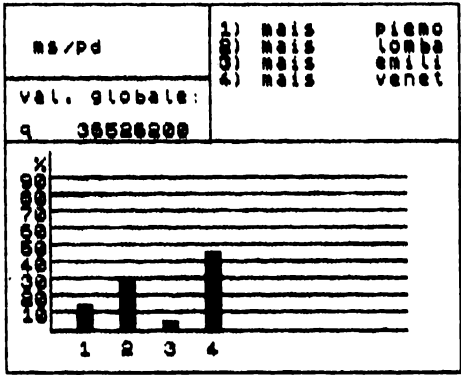
A questo punto il dialogo termina e riprende con il menù principale: nel computer è

stata costruita la tabella dell'istogramma, non resta che farlo disegnare scegliendo l'opzione DISEGNARE ISTO.



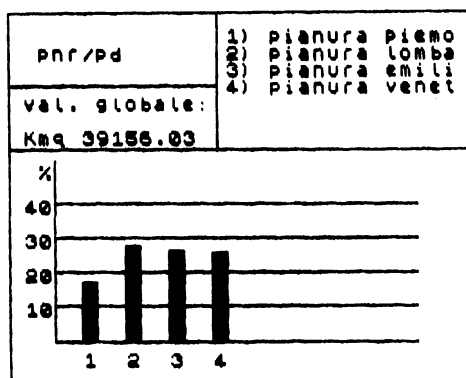
Prima di proseguire nell'analisi sarà opportuno che l'utente provveda a conservare nel cartridge dei dati l'istogramma grn/pd per poterlo poi richiamare all'occorrenza, dopo di che l'analisi geografica può precedere seguendo la linea operativa appena descritta.

Lo studente può allora chiedersi ragione delle differenze quantitative per la produzione complessiva di cereali (grano+mais) e interroga quindi il computer facendogli stampare l'istogramma relativo alle estensioni superficiali delle tre regioni, supponendo che la minor produzione del Piemonte dipenda da una minor superficie regionale: l'istogramma che ottiene è il seguente:



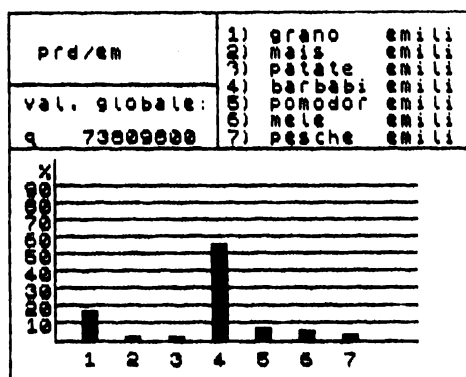
Il risultato non conferma l'ipotesi, si rende quindi necessario formulare una nuova ipotesi e verificarla: la differenza di produzione dipende anche dalla natura del territorio, i cereali sono coltivati generalmente in pianura quindi nel caso specifico il Piemonte dovrebbe risultare poco pianeggiante.

L'istogramma conferma questa seconda ipotesi:



Dagli istogrammi sulla produzione dei cereali emerge un elemento; la modesta produzione di mais in Emilia-Romagna; è ipotizzabile in proposito che il territorio pianeggiante ospiti altre culture prodotte in quantità significative.

L'istogramma sulla *produzione agricola* dell'Emilia-Romagna conferma l'ipotesi evidenziando una elevata produzione di barbabietole

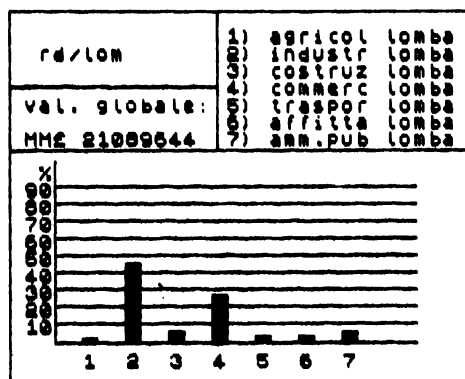


Quanto presentato consente di capire in quale modo sia possibile con il computer operare correlazioni tra dati al fine di verificare ipotesi, non procediamo però oltre nell'analisi in quanto non è scopo di questa pubblicazione approfondire la conoscenza della pianura padana.

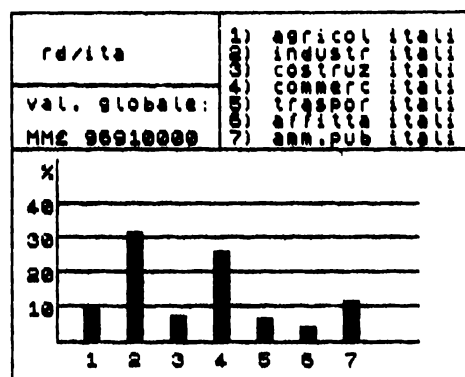
Contributo regionale alla produzione della nazione

La correlazione che si vuole stabilire è tra dati espressi su scala regionale e dati nazionali; per fissare le idee si prende in considerazione la produzione di reddito della Lombardia.

L'istogramma a livello regionale è il seguente



mentre a livello nazionale si ha



Le fonti di reddito più significative risultano in entrambi i casi l'industria ed il commercio; la correlazione intende stabilire quanto il reddito prodotto in Lombardia da queste due attività contribuisce a livello nazionale.

Si costruisce un istogramma per la variabile "Lombardia" con i dati:

industria
commercio

all'atto però di fornire il valore globale si dovrà scegliere l'opzione che consente di prelevare tale valore dalla tabella generale (opzione 3).

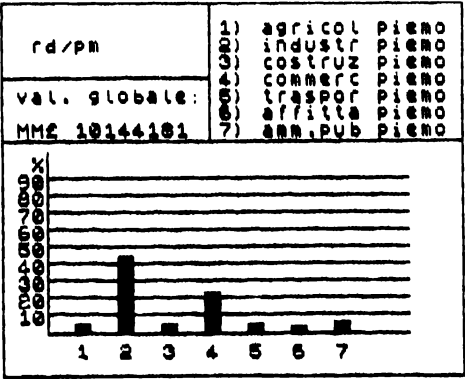
Il computer ci chiederà prima di indicare il nome della nuova variabile cui è riferito il valore globale: nel nostro caso si dovrà rispondere "Italia"; dopo di comunicargli se il valore globale è la somma del valore dei dati per la nuova variabile: nel nostro caso si dovrà rispondere di sì.

Infatti le percentuali per il reddito industriale e commerciale lombardo dovranno essere calcolate sulla somma dei redditi industriale e commerciale di tutta l'Italia.

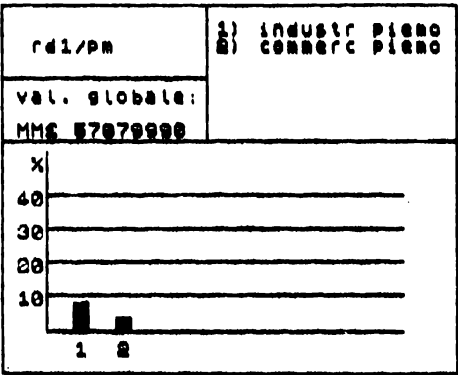
Risultato di questa operazione è l'istogramma desiderato.

È interessante vedere quali risultati produce questa stessa elaborazione quando venga applicata ad altra regione.

Di seguito prenderemo in esame il Piemonte che come appare nell'istogramma ha un reddito distribuito in modo del tutto analogo a quello lombardo, cioè con una netta prevalenza nei settori industriale e commerciale, addirittura per l'industria entrambi raggiungono il 46%.



Una lettura superficiale porterebbe ad affermare che Lombardia e Piemonte contribuiscono al reddito nazionale in uguale misura per il settore industriale, l'elaborazione sopra descritta fornisce un istogramma che contraddice questa ipotesi.



La ragione sta ovviamente nel fatto che il reddito globale della Lombardia è doppio di quello del Piemonte (MM£ 21.089.644 contro MM£ 10.144.181).

Anche in questo esempio emergono le modalità di impiego del programma utility, la cui funzione consiste nel far costruire agli studenti informazioni a partire dai dati operando disaggregazioni e/o accoppiamenti.

In altre parole i dati diventano oggetto di analisi e la loro elaborazione lo stimola a formulare ipotesi e verifiche; al programma sono affidati compiti di calcolo e di rappresentazione grafica, agli studenti, sotto la guida dell'insegnante, spetta:

- Formulare il piano di analisi.
- Provvedere ad impostare e far eseguire al computer le elaborazioni.
- Leggere gli istogrammi e dedurre la convalida o meno delle ipotesi di partenza.

L'uso del programma consente inoltre di stimolare negli studenti la ricerca di nuovi dati e/o precisazioni su quelli di cui già dispongono in Tabella GENERALE, ciò è reso possibile dal fatto che una elaborazione può essere praticabile in quanto in Tabella GENERALE non sono presenti le disaggregazioni dei dati.

In proposito va comunque precisato che la ricerca di dati pronti per l'elaborazione non è compito facile; l'insegnante dovrebbe quindi limitare agli studenti il campo di intervento circoscrivendo la consultazione ai soli libri di testo; l'Atlante di Lavoro e relativo corso di geografia sono in tal senso un esempio significativo di strumenti didattici da cui poter attingere dati.

3.2.5. Il modello per risolvere

Il Menù

La costruzione del programma di utilità viene organizzata con il seguente menù:

```

                                ISTOGRAMMI

PER RICHIAMARE DATI PREMI -> 1
PER INSERIRE DATI PREMI ---> 2
PER DISEGNARE ISTO PREMI --> 3
PER LISTARE DATI PREMI ----> 4
PER CONSERVARE DATI PREMI -> 5
PER CORREGGERE PREMI -----> 6
USCIRE DAL PROGRAMMA PREMI > 7

SCEGLI UN NUMERO POI PREMI ENTER

```

cui corrisponde il programma principale:

```

1 BORDER 4
5 CLS
10 REM ISTOGRAMMI
20 PRINT AT 0,11;"ISTOGRAMMI"
40 PRINT AT 4,1;"PER RICHIAMAR
E DATI PREMI -> 1"
50 PRINT AT 6,1;"PER INSERIRE
DATI PREMI ---> 2"
60 PRINT AT 8,1;"PER DISEGNARE
ISTO PREMI --> 3"
70 PRINT AT 10,1;"PER LISTARE
DATI PREMI ----> 4"
80 PRINT AT 12,1;"PER CONSERVA
RE DATI PREMI -> 5"
90 PRINT AT 14,1;"PER CORREGGE
RE PREMI -----> 6"
95 PRINT AT 16,1;"USCIRE DAL P
ROGRAMMA PREMI > 7"
100 PRINT AT 20,0; FLASH 1;"SCE
GLI UN NUMERO POI PREMI ENTER"
110 PLOT 0,0; DRAW 255,0; DRAW
0,175; DRAW -255,0; DRAW 0,-175

```

```

120 INPUT K
130 CLS
140 GO SUB 1000*K
150 INPUT "ANCORA? (S/N) ";V
$
180 IF V$="S" THEN GO TO 130
190>CLS
200 GO TO 5

```

dove le istruzioni 120 e 140 provvedono all'invio nei sottoprogrammi.

La gestione del programma prevede che vengano in primo luogo richiamati dei dati dal microdrive oppure inseriti direttamente dati dopo di che sarà possibile utilizzare le altre operazioni, diversamente sul video comparirà il messaggio

2. variabile not found

per comunicare all'utente che gli indirizzi di memoria sono vuoti.

Per richiamare dati

Come già detto bisogna provvedere a caricare in memoria i dati necessari alla costruzione degli istogrammi in quanto è previsto che il programma di utilità nel suo insieme venga caricato oppure salvato su cartridge senza dati; il motivo è duplice:

- il programma senza dati occupa meno memoria sul nastro,
- il programma deve essere utilizzato con insiemi di dati che possono variare di volta in volta.

Vanno previste due tipologie di dati da richiamare, e cioè:

- la tabella dei dati relativa a tutti gli istogrammi disegnati su uno stesso riferimento,
- la Tabella GENERALE contenente tutti i dati disponibili per la comparazione.

Inoltre qualora si intenda impostare una nuova Tabella GENERALE di comparazione dovrà essere consentito provvedervi direttamente alla tastiera.

A lavoro terminato dovrà essere previsto un ritorno al Menù principale.

Prima di esaminare in dettaglio le modalità di richiamo delle tabelle si riporta di seguito il menù secondario appena descritto:

```

          RICHIAMARE DATI

per richiamare la tabella
dell'istogramma premi -----> 1

per richiamare la tabella
GENERALE premi -----> 2

per impostare la tabella
GENERALE premi -----> 3

per tornare al MENU premi -> 0

SCEGLI UN NUMERO POI PREMI ENTER

```

ed il relativo blocco di programma

```

1000>REM RICHIAMARE DATI
1010 PRINT TAB 9,"RICHIAMARE DATI"
1020 PRINT AT 4,1:"per richiamar
e la tabella dell'istogramma
premi ----> 1"
1030 PRINT AT 8,1:"per richiamar
e la tabella GENERALE prem
i ----> 2"
1040 PRINT AT 12,1:"per impostar
e la tabella GENERALE pr
emi ----> 3"
1050 PRINT AT 17,1:"per tornare
al MENU premi -> 0"
1060 PLOT 0,0 DRAW 255,0 DRAW
0,175 DRAW 255,0 DRAW 0,-175
1065 PRINT AT 20,0 FLASH 1,"SCE
GLI UN NUMERO DEI PREMI ENTER"
1070 INPUT "CLS
1080 IF 0=0 THEN GO TO 5
1090 GO TO 1000+100*n

```

dove le istruzioni 1070 e 1090 sono responsabili del rinvio ai singoli blocchi di istruzioni previsti dal menù secondario.

Richiamare la tabella dell'istogramma

La tabella è destinata a contenere:

- il nome di ogni dato di cui si disegna l'istogramma,
- il valore percentuale di ogni dato.

Inoltre per le ragioni dette sopra è necessario affiancare alla tabella il valore globale dei dati.

Parallelamente a questa tabella ne può essere costruita una delle stesse dimensioni che invece di contenere i valori percentuali contiene i corrispondenti valori assoluti, l'utilizzo di questa nuova tabella, non necessaria al disegno degli istogrammi, è destinato alla traduzione diretta delle percentuali in valori assoluti.

Infine per ragioni che saranno più chiare in seguito, quando ci occuperemo dell'inserimento dei dati, la serie dei dati si completa con quella relativa al numero di elementi contenuti in tabella.

La struttura delle tabelle suggerisce l'impiego delle matrici di memoria che come noto vengono predisposte con la parola-chiave DIM, ora si deve sapere che lo Spectrum ammette l'apertura in RAM di una matrice di memoria completa di dati in essa collocati eseguita con il caricamento da cartridge, senza che debba in precedenza essere scritto un programma che abilita il computer a questa operazione.

Il formato dell'istruzione di caricamento è simile a quello usuale solo che di seguito va ora collocata la parola-chiave DATA, l'indirizzo della matrice di memoria che si intende aprire e una coppia di parentesi, cioè:

LOAD * "m":

NUM DRIVE

 ; "nome file" DATA

indiriz. matrice memoria

 ()

Per poter utilizzare la procedura appena descritta basta quindi trasferire la tabella dell'istogramma nelle corrispondenti matrici di memoria: la soluzione proposta adotta quattro differenti matrici, e precisamente:

- 1 la matrice alfanumerica s\$ in cui sono inseriti i nomi dei dati e le unità di misura,
- 2 la matrice alfanumerica w\$ che contiene il nome dell'istogramma e il numero complessivo di dati,
- 3 la matrice numerica q dove sono presenti i valori assoluti e il valore globale cui sono riferiti i dati,

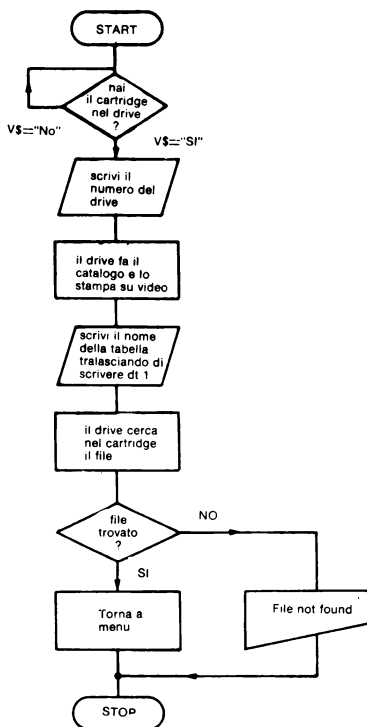
4 la matrice numerica u che ospita i valori percentuali.

Le modalità di riempimento delle matrici saranno spiegate quando ci si occuperà del sottoprogramma INSERIRE DATI.

Questa è la lista delle istruzioni per richiamare la tabella dell'istogramma:

```
1100>REM richiamare la tabella d  
ell'istogramma  
1110 INPUT "hai la cartridge nel  
drive? s/n";v$  
1115 IF v$<>"s" THEN GO TO 1110  
1120 INPUT "scrivi il numero del  
drive";x  
1125 DAT x  
1130 INPUT "scrivi il nome della  
tabella tralasciando di scriver  
e dt1";v$  
1140 LOAD "*"m";x;v$+"dt1" DATA s  
$()  
1150 LOAD "*"m";x;v$+"dt2" DATA q  
()  
1160 LOAD "*"m";x;v$+"dt3" DATA u  
()  
1170 LOAD "*"m";x;v$+"dt4" DATA w  
$()  
1180 LET t=VAL w$(1)  
1190 RETURN
```

Il diagramma di flusso mostra ciò che l'utente si vede domandare dal computer e come evolve il colloquio in base alle risposte che fornisce.



Il cartridge richiesto all'utente non è lo stesso su cui è inciso il programma di utilità, ma è quello destinato a contenere tutti e solo i dati. Per il modo in cui è strutturato il programma ogni tabella compare nel catalogo con quattro nomi come si precisa di seguito.

Un esempio di catalogo è il seguente dove per altro compare anche la tabella generale riconoscibile dalla desinenza Gdt.

```
istodati
It-ecGdt1
It-ecGdt2
It-ecGdt3
It-ecGdt4
grn/Pddt1
grn/Pddt2
grn/Pddt3
grn/Pddt4

ns/pd dt1
ns/pd dt2
ns/pd dt3
ns/pd dt4
```

A completamento del programma vanno date due spiegazioni.

La prima riguarda appunto le desinenze dt1, dt2, dt3, dt4; averle utilizzate ha significato ridurre ad una sola la richiesta verso l'utente, che in caso contrario per richiamare i quattro file avrebbe dovuto scrivere quattro nomi differenti.

La seconda si riferisce all'istruzione 1180 con la quale si pone nell'indirizzo t il numero di dati di cui è composta la matrice; poiché la matrice w\$ è alfanumerica in quanto vi è contenuto anche il nome dell'istogramma, per convertire il carattere-cifra in numero è stato necessario utilizzare la funzione VAL.

Richiamare la tabella generale

La procedura non si differenzia da quella appena esposta, quindi ci limitiamo a presentare la trasposizione della tabella in matrici di memoria.

Come si è detto si tratta di una tabella a doppia entrata in cui ci sono solo valori assoluti messi in corrispondenza con i nomi delle VARIABILI e dei DATI; di nuovo la soluzione proposta adotta quattro differenti matrici così utilizzate:

- 1 matrice alfanumerica b\$ dove sono contenuti i nomi delle VARIABILI,
- 2 matrice alfanumerica a\$ in cui sono inseriti i nomi dei DATI e le unità di misura,
- 3 matrice numerica bidimensionale-a-che ospita tutti i valori assoluti,
- 4 matrice alfanumerica y\$ dove, oltre al nome della tabella, sono contenuti il numero dei DATI e il numero delle VARIABILI di cui è fatta la matrice: a.

La lista delle istruzioni è la seguente:

```
1200>REM richiamare la tabella g
generale
1210 INPUT "hai la cartridge nel
drive? s/n/v$":v$
1215 IF v$<>"s" THEN GO TO 1210
1220 INPUT "scrivi il numero del
drive":x
1225 CAT x
1230 INPUT "scrivi il nome della
tabella lasciando di scriver
e Gdt1":v$
1240 LOAD "*"m"/x/v$+"Gdt1" DATA
b$(1)
1250 LOAD "*"m"/x/v$+"Gdt2" DATA
a$(1)
1260 LOAD "*"m"/x/v$+"Gdt3" DATA
a$(1)
1270 LOAD "*"m"/x/v$+"Gdt4" DATA
y$(1)
1280 LET v=VAL y$(1): LET d=VAL
y$(2)
1290 RETURN
```

Impostare la tabella generale

Questa operazione permette all'utente di generare una propria tabella indipendentemente da quelle già contenute nel cartridge

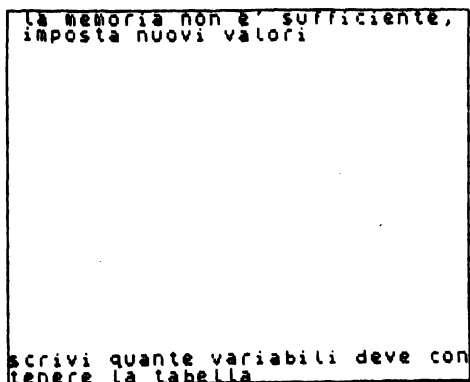
La lista delle istruzioni che proponiamo:

```
1300>REM impostare tabella generale
1310 INPUT "scrivi quante variabili deve contenere la tabella";v
1320 INPUT "scrivi quanti dati deve contenere la tabella";d
1325 IF d*25+v*15+d*v*5>25000 THEN PRINT FLASH 1;"la memoria non e' sufficiente, imposta nuovi valori";GO TO 1310
1330 DIM a$(d,25): DIM b$(v,15): DIM a(d,v): DIM y$(3,15)
1335 LET y$(1)=STR$ v: LET y$(2)=STR$ d
1340 CLS: INPUT "scrivi il nome della tabella ";y$(3): PRINT y$(3)
1350 PRINT "la tabella generale e' stata costruita per contenere:","v;" variabili","d;" dati per"," variabile"
1360 RETURN
```

merita pochi commenti in quanto le matrici di memoria che vengono dimensionate alla riga 1330 sono quelle già descritte nel precedente paragrafo.

Il dimensionamento pone il problema delle quantità di memoria che il computer è in grado di mettere a disposizione; il computer alla riga 1325 si preoccupa di verificare che l'occupazione di memoria non superi i 25 Kbyte, quindi tenuto conto che ogni carattere alfanumerico necessita di 1 byte e ogni numero di 5 byte pone la condizione che $d*25+v*15+d*v*5$ non superi il valore 25.000.

Qualora i limiti fossero superati sul video compare il messaggio lampeggiante per informare l'utente che:



la memoria non e' sufficiente,
imposta nuovi valori

scrivi quante variabili deve contenere la tabella

Per fissare gli ordini di grandezza si consideri che impostando la tabella per 20 variabili e 70 dati la memoria occupata non supera i 9Kbyte.

Infine va notato come l'istruzione 1335 provvede ad inserire i valori numerici per i totali di VARIABILI e DATI nella memoria alfanumerica y\$; allo scopo il BASIC mette a disposizione la funzione STR\$ il cui comportamento è opposto a quello della funzione VAL.

Per inserire dati

Questo sottoprogramma occupa un ruolo centrale in quanto provvede non solo a ricevere dati dall'utente, ma anche ad eseguire tutte le elaborazioni necessarie a trasferire dati dalla tabella generale alla tabella dell'istogramma, come risulta chiaramente dalle opzioni offerte dal menù secondario:

```

INSEIRE DATI

per inserire dati direttamen
te in tabella di istogramma
premi -----> 1

per inserire dati in
tabella GENERALE premi ----> 3

per inserire dati in tabella
di istogramma prelevandoli
da tabella GENERALE premi -> 5

per tornare al MENU premi -> 0

SCEGLI UN NUMERO POI PREMI ENTER

```

Elemento comune alle tre operazioni è la predisposizione dei dati ad essere tradotti in valori percentuali rappresentabili con istogrammi, di seguito illustreremo nell'ordine i relativi sottoprogrammi.

La lista delle istruzioni che realizza il menù secondario è standard e non merita alcun commento:

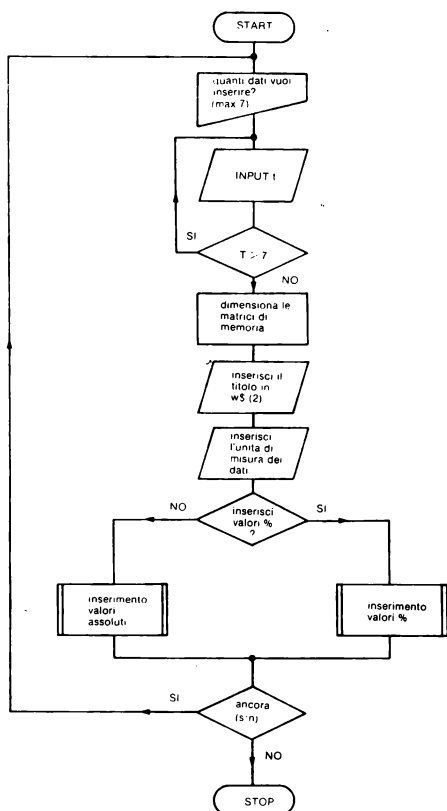
```

2000>REM INSEIRE DATI
2010 PRINT TAB 10;"INSEIRE DATI
"
2020 PRINT AT 4,1;"per inserire
dati direttamen te in tabella
di istogramma      premi -----
-----> 1"
2030 PRINT AT 8,1;"per inserire
dati in            tabella GENER
ALE premi ----> 3"
2040 PRINT AT 12,1;"per inserire
dati in tabella   di istogramm
a prelevandoli    da tabella G
ENERALE premi -> 5"
2050 PRINT AT 17,1;"per tornare
al MENU premi -> 0"
2060 PRINT AT 20,0; FLASH 1;"SCE
GLI UN NUMERO POI PREMI ENTER"
2070 PLOT 0,0 DRAW 255,0; DRAW
0,175; DRAW -255,0; DRAW 0,-175
2080 INPUT n: CLS
2085 IF n=0 THEN GO TO 5
2090 GO TO 2000+100*n

```

Per inserire dati direttamente in tabella di istogramma

È questo il caso in cui l'utente in possesso di propri dati, intende visualizzarli con un istogramma; la procedura si articola in vari passaggi cui corrispondono una serie di alternative poste all'utente che dovrà scegliere in funzione dei dati di cui dispone. Vediamo nel diagramma a blocchi descritta l'intera procedura per questo sottoprogramma:



cui corrispondono le seguenti istruzioni BASIC:

```

2100>REM inserire dati direttamente
2110 INPUT "quanti dati vuoi inserire? (massimo consentito 7)"; t: IF t>7 THEN GO TO 2110
2115 DIM s$(t,25): DIM q(t+1): DIM u(t): DIM w$(2,15)
2118 LET w$(1)=STR$ t
2120 INPUT "scrivi il titolo dell'istogramma"; w$(2): PRINT TAB 3; w$(2)
2130 INPUT "scrivi l'unità di misura comune a tutti i dati che inserirai "; d$: FOR i=1 TO t: LET s$(i,20)=d$: NEXT i
2140 INPUT "inserisci dati in % (s/n)"; v$: IF v$="s" THEN GO TO 2240
2145 REM inserire valori assoluti
2230>RETURN
2240 REM inserire valori percentuali
2250 INPUT "scrivi il valore globale dei dati (Pari al 100%)";
  
```

Alle righe 2145 e 2240 iniziano i due sottoprogrammi che l'utente sceglie in base alla tipologia dei dati di cui è in possesso: valori assoluti oppure valori percentuali.

Prima di passare a questi due sottoprogrammi merita descrivere come sono strutturate dalla riga 2115 le matrici di memoria destinate a ricevere i dati, delle quali abbiamo già detto in parte nel paragrafo dedicato al sottoprogramma RICHIAMARE DATI.

Memoria s\$: è una memoria alfanumerica che contiene t dati di 25 caratteri ciascuno di cui i primi 19 saranno occupati dal nome del dato e i restanti 6 caratteri saranno destinati all'unità di misura (cfr riga 2130).

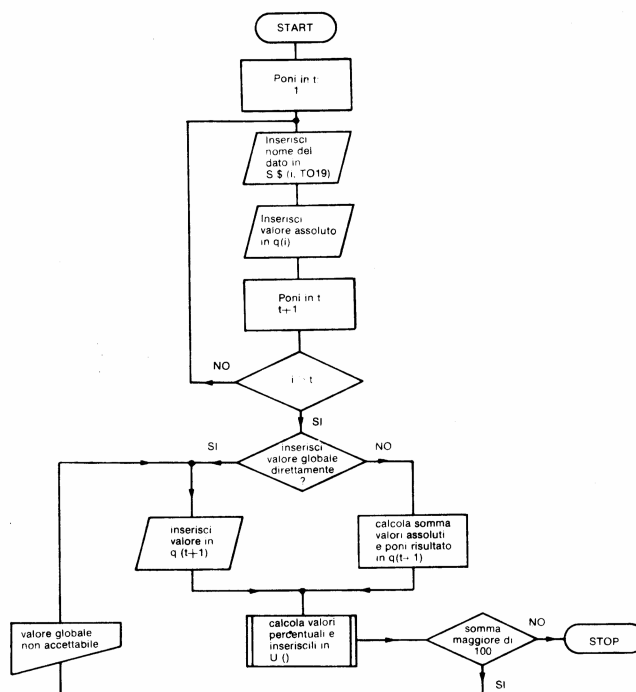
- Memoria q: è una matrice numerica con $t+1$ indirizzi, i primi t conterranno ciascuno il valore assoluto del t -esimo dato, nel $(t+1)$ -esimo verrà invece inserito il valore globale dei dati.
- Memoria u: è anch'essa una matrice numerica con t indirizzi in ciascuno dei quali verrà collocato il valore percentuale del rispettivo t -esimo dato.
- Memoria w\$: è una matrice alfanumerica con due indirizzi di memoria di 15 caratteri ciascuno; è destinata a contenere due dati, l'uno alfanumerico (il nome dell'istogramma, cfr riga 2120) l'altro numerico (il numero di dati, cfr riga 2118).

Per inserire un valore numerico in una matrice alfanumerica è necessario usare una funzione adeguata: la funzione è STR\$ che ha giusto lo scopo di trasformare un numero in cifre leggibili come singoli caratteri.

Passiamo ora ai sottoprogrammi di inserimento valori; al termine di entrambi deve essere raggiunto un duplice obiettivo: nelle memorie devono essere contenuti sia i valori assoluti sia i valori percentuali nonostante i sottoprogrammi ricevano dall'utente o valori assoluti o valori percentuali, allo scopo provvede il valore globale.

Inserisce valori assoluti

La procedura risolutiva è descritta dal seguente diagramma di flusso;



Nel calcolo del valore percentuale si è usata la formula:

$$\text{INT} (q(i)/q(t+1)*10.000)/100$$

che consente di conservare del rapporto due sole cifre decimali.

Esempio se $q(1) = 238$

$q(t+1) = 738$

```
q(1)/q(t-1)=0.32249323  
INT (0.32249323*10000)/100=32.24
```

La lista delle istruzioni è pertanto la seguente dove la riga 2225 controlla che la somma dei valori percentuali calcolati non superi il 100% contrariamente richiede un nuovo valore globale:

```
2145>REM inserire valori assoluti  
2150 FOR i=1 TO t  
2160 INPUT "scrivi il nome del dato ";s$(i,1 TO 19)  
2170 PRINT " ";s$(i,1 TO 19)  
2180 INPUT "scrivi il valore del dato ";q(i); PRINT TAB 8;  
s$(1,20 TO 19);q(i); NEXT i  
2190 INPUT "vuoi inserire il valore globale direttamente? (s/n) ";v$  
IF v$="s" THEN INPUT "scrivi il valore globale ";q(t+1); GO TO 2210  
2200 LET q(t+1)=0; FOR i=1 TO t:  
LET q(t+1)=q(t+1)+q(i); NEXT i  
2210 PRINT "valore globale dei dati: ";TAB 8;s$(1,20 TO 19);q(t+1)  
2220 FOR i=1 TO t: LET u(i)=INT (q(i)/q(t+1)*10000)/100; NEXT i  
2225 LET h=0; FOR i=1 TO t: LET h=h+u(i); NEXT i: IF h>100 THEN  
PRINT "valore globale non accettabile";  
"inserire nuovo valore"; GO TO 2190  
2230 RETURN
```

Inserisce valori percentuali

In questo caso è bene che il valore globale sia subito inserito per evitare che i valori percentuali inseriti rimangano privi di un termine di riferimento.

Senza ricorrere al diagramma di flusso, dalla lettura del programma si deduce che la procedura richiede l'inserimento dei dati poi (cfr riga 2270) controlla che la somma dei valori inseriti non superi il 100%, contrariamente rifiuta i dati in blocco e li richiede tutti di nuovo. Infine provvede con la riga 2280 al calcolo dei valori assoluti ed al relativo inserimento nelle memorie q().

La lista delle istruzioni è:

```
2240>REM inserire valori percentuali  
2250 INPUT "scrivi il valore globale dei dati (pari al 100%) ";q(t+1); PRINT "valore globale dei dati: ";TAB 8;s$(1,20 TO 19);q(t+1)  
2260 FOR i=1 TO t: INPUT "scrivi il nome del dato ";s$(i,1 TO 19); PRINT " ";s$(i,1 TO 19); INPUT "scrivi il valore in % del dato ";u(i); PRINT TAB 8;u(i);" %"; NEXT i  
2270 LET h=0; FOR i=1 TO t: LET h=h+u(i); NEXT i: IF h>100 THEN  
PRINT "valori non accettabili";  
"inserire valori % corretti"; GO TO 2260  
2280 FOR i=1 TO t: LET q(i)=q(t+1)*u(i)/100; NEXT i  
2290 RETURN
```

Per inserire dati in tabella GENERALE

Come si è già detto nel paragrafo che si occupa del sottoprogramma RICHIAMARE DATI, la tabella GENERALE è strutturata come tabella a doppia entrata i cui lati sono definiti dalle VARIABILI e dai DATI. Pertanto il numero di dati da inserire è dipendente dal numero di variabili per le quali si intendono inserire quei dati.

Per fissare un modello cartaceo di riferimento continueremo a far ricorso alle tabelle dei confronti regionali d'Italia dell'Atlante di Lavoro di Giulio Mezzetti.

In primo luogo si dovrà dichiarare per quante variabili si vogliono assegnare dati, cioè nel nostro caso per quante regioni d'Italia, poi si passerà a definire quanti sono i dati che inseriremo per ogni variabile: i dati sono gli stessi per tutte le variabili previste.

In altre parole se voglio inserire i dati di produzione di mais e grano per la Lombardia, per il Piemonte e per il Veneto, basterà indicare che mi necessitano 3 variabili e 2 dati; sulla base delle seguenti istruzioni:

```
2300>REM inserire dati in tabell
a generale
2310 INPUT "quante variabili ti
occorrono? " f
2315 PRINT f;TAB 10;"VARIABILI"

2320 DIM c$(f,15): FOR i=1 TO f:
INPUT "scrivi il nome della var
iabile " c$(i): PRINT c$(i)
NEXT i
2330 INPUT "quanti dati inserisc
i per ogni variabile? " g
2335 PRINT g;TAB 12;"DATI"
2340 INPUT "inserisci dati in %
(s/n) " v$: IF v$<>"s" THEN
N GO TO 2367

2343>REM inserisci valori in %
2367>REM inserisci valori assolu
ti
```

Nel programma la memoria c\$ è dimensionata per contenere gli f nomi delle variabili.

Prima di poter comunicare il nome dei dati, cioè grano e mais bisogna scegliere tra inserire i *valori in percentuale* oppure i *valori assoluti*. L'alternativa comporta che venga eseguito il seguente sottoprogramma:

```
2343>REM inserisci valori in %
2345 DIM e$(g,25): INPUT "scrivi
l'unità di misura del valore
globale" e$(1,20 TO ): FOR i=1
TO g: LET e$(i,20 TO )=e$(1,20 T
O ): NEXT i
2350 INPUT "scrivi il valore glo
bale dei dati (pari al 100%)" z:
PRINT "valore globale dei d
ati: " ;TAB 8:e$(1,20 TO ),z
LET z=z/100
2355 FOR i=1 TO g: INPUT "scrivi
il nome del dato" e$(i, TO 19):
PRINT e$(i, TO 19): NEXT i
2360 INPUT "va bene? (s/n) " v$
IF v$="n" THEN CLS : GO TO 230
0
2365 GO TO 2380
```

oppure quest'altro:

```

2367>REM inserisci valori assolu
ti
2370 DIM e$(9,25): LET z=1: FOR
i=1 TO 9: INPUT "scrivi il nome
del dato": e$(i, TO 19): PRINT e$
(i, TO 19): INPUT "scrivi l'uni
ta di misura": e$(i,20 TO ): PRI
NT e$(i,20 TO ): NEXT i
2375 INPUT "va bene? (s/n) ": v$
: IF v$="n" THEN CLS : GO TO 230
0

```

Entrambi i programmi dimensionano una memoria e\$ affinché possa contenere i g nomi dei dati, unità di misura comprese, si differenziano invece per le modalità di inserimento.

Infatti il programma che inserisce i valori percentuali in primo luogo richiede l'unità di misura per il valore globale (100%).

Invece il programma che inserisce i valori assoluti chiede subito il nome dei dati, poi l'unità di misura, che viene associata al dato appena inserito e così via sino al termine man mano che compaiono sul video le scritte.

La ragione delle differenti richieste sta nella diversa natura dei valori che saranno poi forniti dall'utente:

— quelli percentuali sono numeri puri, doverli tradurre in valori assoluti dotati di unità di misura come richiesto per tutti i dati presenti in Tabella GENERALE è possibile solo a condizione di conoscere in anticipo il valore globale cui fanno riferimento le percentuali e l'unità di misura con cui tale valore è espresso;

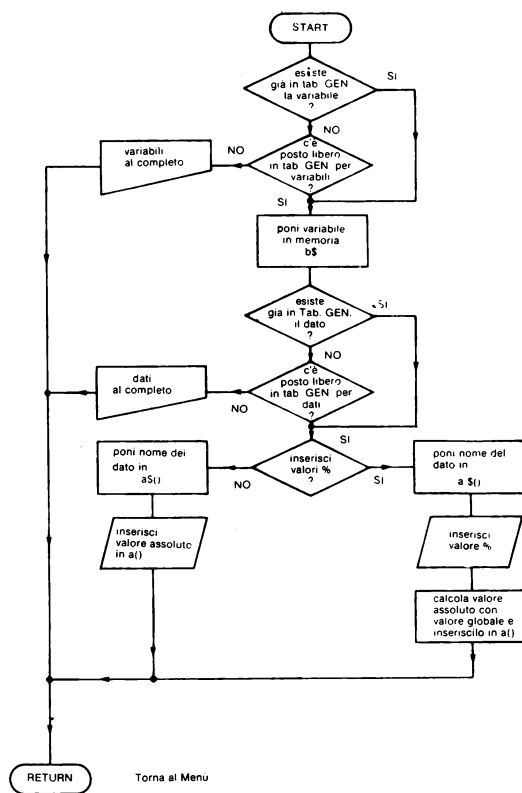
— i valori assoluti sono già pronti per l'inserimento in Tabella GENERALE, non è cioè necessaria alcuna elaborazione, bastano solo: nome, unità di misura e, al tempo opportuno, valore assoluto.

Prima di procedere con il resto del sottoprogramma di inserimento dati in Tabella GENERALE vale la pena di osservare che con la procedura appena descritta non è consentito selezionare due o più variabili e due o più dati e poi inserire per ogni variabile meno dati di quelli selezionati.

In altre parole se come variabili si scelgono Lombardia e Veneto, e come dati mais e grano poi non si può inserire solo il valore del mais per la Lombardia e non quello del grano e viceversa per il veneto solo il valore del grano e non quello del mais.

A questo scopo si perviene inserendo i dati in due tempi successivi: prima Lombardia e mais, poi al termine, dopo essere tornato al menù e ritornato al sottoprogramma: Veneto e grano.

Dopo aver così impostato le procedure di inserimento si arriva all'operazione vera e propria che si articola come descritto nel diagramma relativo alla routine di INSERIMENTO VALORI che limitiamo per semplicità ad una sola variabile e un solo dato; la generalizzazione a più variabili e più dati richiede la serie di iterazioni concatenate presenti nella lista delle istruzioni.



Nella lista delle istruzioni sotto riprodotta si nota che la condizione di inserimento di valori percentuali oppure no, è stabilita in base al valore contenuto nella memoria z che nei precedenti sottoprogrammi si era provveduto a riempire in un caso con il valore globale/100 e nell'altro con l'unità.

```

2380>FOR i=1 TO f: REM routine d
i inserimento valori
2385 FOR h=1 TO v: REM cerca VAR
IABILE
2390 IF c$(i)=b$(h) THEN GO TO 2
425
2405 NEXT h
2410 FOR h=1 TO v: REM cerca pos
to per VARIABILE
2415 IF CODE b$(h)=32 THEN GO TO
2425
2420 NEXT h: PRINT "VARIABILI
della tabella GENERALE al complet
o": RETURN
2425 LET b$(h)=c$(i): PRINT TAB
(h," ) ",b$(h)
2430 FOR l=1 TO g
2435 FOR j=1 TO d: REM cerca DAT
2440 IF a$(l, TO 19)=a$(j, TO 19
) THEN GO TO 2463
2445 NEXT j
2450 FOR j=1 TO d: REM cerca pos
to libero per DATO
2455 IF CODE a$(j)=32 THEN GO TO
2463

```

```

2460 NEXT j: PRINT "DATI della
tabella GENERALE al completo":
RETURN
2463 IF z<>1 THEN LET a$(j)=e$(l)
: INPUT +a$(j), TO 19): " % "a(j
,h): LET a(j,h)=a(j,h)*z: REM in
serisce valore assoluto calcolat
o con valore percentuale in memo
ria
2465 IF z=1 THEN LET a$(j)=e$(l)
: INPUT +a$(j),a(j,h): REM inser
isce valore in memoria a(j)
2470 PRINT j: " "a$(j, TO 19):
TAB 8,a$(j,20 TO j,a(j,h):
2475 NEXT l
2480 NEXT i
2490 RETURN

```

Inserire dati prelevati da Tabella GENERALE

Il sottoprogramma che descriviamo di seguito svolge un ruolo fondamentale nell'adattare alle esigenze di discipline scolastiche che fanno largo uso di dati, l'intero programma.

La funzione cui è destinato consiste nel trasferire valori dalla Tabella GENERALE alla tabella per gli istogrammi, dove si collocano gli elementi utili alla rappresentazione grafica.

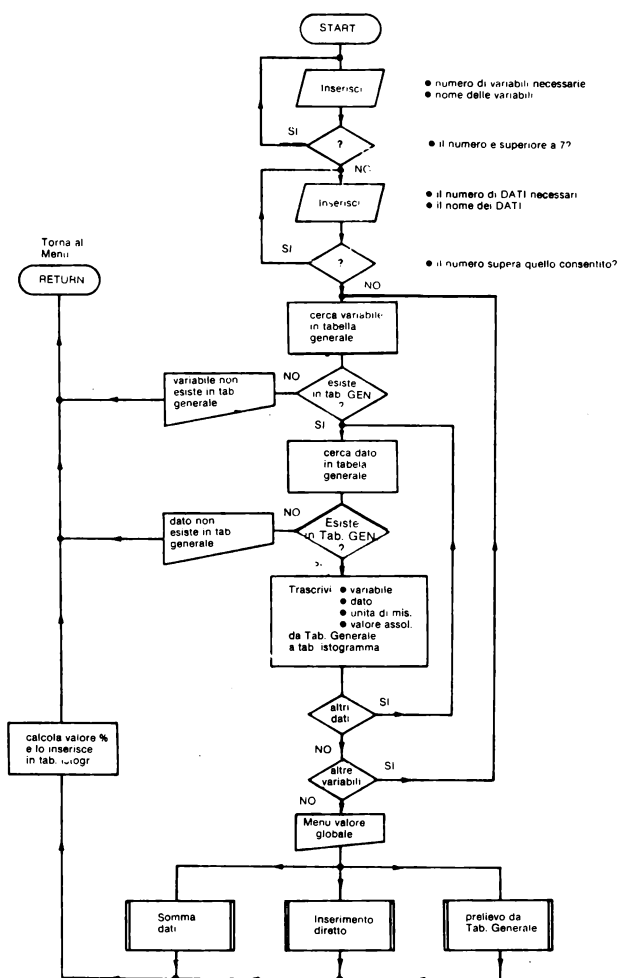
L'elaborazione di dati consentita dal sottoprogramma permette di operare confronti e correlazioni tra tutti i valori assoluti presenti nella Tabella GENERALE in quanto possono essere trasferiti per gruppi alla tabella dell'istogramma con queste possibili associazioni:

- un dato per un insieme di variabili,
- un insieme di dati per un insieme di variabili,
- un insieme di dati per una variabile.

La consistenza numerica di questi insiemi è limitata dal numero di istogrammi rappresentabili sullo stesso sistema di riferimento, che come descriveremo nel prossimo paragrafo, è per questo programma di sette istogrammi.

Il fatto che i dati trasferiti sono tutti espressi in valore assoluto mentre la rappresentazione è per valori percentuali permette di agire su un ulteriore elemento per fare comparazioni e cioè scegliere il valore globale su cui calcolare le percentuali.

L'intera procedura è descritta sommariamente dal seguente diagramma.



Passiamo ora in dettaglio le singole parti del sottoprogramma.

Inserimento variabili e dati

Con il seguente gruppo di istruzioni si provvede a comunicare nelle memorie di servizio c\$ ed e\$ il nome delle variabili e dei dati controllando che non siano più di sette gli istogrammi che si producono dal loro accoppiamento; se tale valore è superato il computer fa ricominciare la procedura dalla riga 2530.

La memoria c\$, per le variabili, e la e\$, per i dati, sono strutturate come matrici bidimensionali per facilitare le operazioni di ricerca all'atto dei confronti che dovranno essere fatti con le corrispondenti memorie della Tabella GENERALE.

Infine la riga 2580, una volta acquisiti tutti gli elementi utili dimensiona le memorie della tabella dell'istogramma usando gli stessi indirizzi di memoria usati nel sottoprogramma INSERIRE DATI DIRETTAMENTE scritto a partire dalla riga 2100.

Unica eccezione la memoria p che in questa circostanza svolge il ruolo di contatore per l'inserimento del nome dei DATA nella memoria s\$:

```

2500>REM inserire dati prelevati
da tabella GENERALE
2510 DIM w$(2,15)
2520 INPUT "scrivi il titolo del
l'istogramma";w$(2): PRINT TAB 3
;w$(2)
2530 INPUT "quante variabili ti
occorrono? (massimo consentito
7)";f: IF f>7 THEN GO
TO 2530
2535 PRINT f;TAB 10;"VARIABILI"
2540 DIM c$(f,15): FOR i=1 TO f:
INPUT "scrivi il nome della var
iabile ";c$(i): PRINT c$(i)
NEXT i
2550 INPUT "quanti dati inserisc
i per ogni variabile? (max cons
entito 7)";INT (7/f);g: IF g>I
NT (7/f) THEN GO TO 2550
2560 PRINT g;TAB 12;"DATI"
2570 DIM e$(g,25): FOR i=1 TO g:
INPUT "scrivi il nome del dato"
;e$(i, TO 19): PRINT e$(i, TO 19)
NEXT i
2580 LET t=f*g: DIM s$(t,25): DI
M q(t+1): DIM u(t): LET w$(1)=ST
R$(t): LET p=1

```

Routine di inserimento valori

Il programma provvede con cicli iterativi a fare il confronto per ordine di inserimento delle variabili richieste con quelle presenti in Tabella GENERALE quindi per ogni variabile trovata procede ad analogo ricerca per tutti i dati richiesti.

Se la ricerca sulla variabile e sul dato dà risultato positivo l'istruzione della riga 2690 provvede al trasferimento di quella variabile, di quel dato e del valore corrispondente nella tabella dell'istogramma:

```

2590>FOR i=1 TO f: REM routine d
i inserimento valori
2600 FOR h=1 TO v: REM cerca VAR
IABILE
2610 IF c$(i)=b$(h) THEN GO TO 2
640
2620 NEXT h
2630 PRINT "la VARIABILE ";c$(i)
;" non esiste nella tabella GENE
RALE": RETURN
2640 FOR l=1 TO g
2650 FOR j=1 TO d: REM cerca DAT
O
2660 IF e$(l, TO 19)=a$(j, TO 19)
THEN GO TO 2690
2670 NEXT j
2680 PRINT "il DATO ";e$(l, TO 1
9);" non esiste nella tabella GE
NERALE": RETURN
2690 LET s$(p, TO 10)=a$(j): LET
s$(p,12 TO 19)=b$(h): LET s$(p,
20 TO )=c$(i,20 TO ): LET q(p)=a
(j,h): REM trasferisce VARIABILE
-DATO-VALORE in tabella istogram
ma
2700 PRINT p;" ";s$(p, TO 19);T
AB 8;s$(p,20 TO );q(p): LET p=
p+1
2710 NEXT l
2720 NEXT i
2725 PRINT FLASH 1:" PREMI UN T
ASTO PER PROSEGUIRE ": PAUSE 0

```

Se o la variabile o il dato non sono nella Tabella GENERALE sul video compare:

```
      prova
2      VARIABILI
lombardia
piemonte
1      DATI
grano
la VARIABILE lombardia non
esiste nella tabella GENERALE
ancora? (s/n)
```

Inoltre il trasferimento del nome della variabile e del dato è stato progettato in modo che risulti in s\$ un nome composto variabile-dato, come è possibile osservare sul video al termine di una operazione di inserimento dati da Tabella GENERALE a tabella dell'istogramma.

```
      cereali/padana
3      VARIABILI
lombardia
emilia-romagna
veneto
2      DATI
grano
mais
1) grano q      lombardi
                9203000
2) mais  q      lombardi
                11373500
3) grano q      emilia-r
                13488700
4) mais  q      emilia-r
                2252000
5) grano q      veneto
                7970900
6) mais  q      veneto
                17100600
      PREMI UN TASTO PER PROSEGUIRE
```

Va detto che la scrittura non sta tutta sul video contemporaneamente per effetto dello scroll che opera una volta che lo schermo è riempito.

Il PAUSE 0 della riga 2725 provoca l'arresto momentaneo del programma che viene riavviato non appena si preme un tasto, come suggerisce la scritta lampeggiante (FLASH 1).

Scelta del valore globale

Il prosieguo del programma propone una scelta da menù

```
VALORE GLOBALE DATI

se valore globale e' somma
dei dati inseriti premi ----> 1

se inserisci valore globale
direttamente premi -----> 2

se prelevi valore globale da
tabella GENERALE premi ----> 3

SCEGLI UN NUMERO POI PREMI ENTER
```

alla cui scrittura provvedono le istruzioni

```
2730>REM scelta del valore globa
le
2731 CLS : PRINT AT 0,6;"VALORE
GLOBALE DATI"
2735 PRINT AT 6,1;"se valore glo
bale e' somma dei dati inse
riti premi ----> 1"
2740 PRINT AT 10,1;"se inserisci
valore globale direttamente
premi -----> 2"
2745 PRINT AT 14,1;"se prelevi v
alore globale da tabella GENE
RALE premi ----> 3"
2750 PRINT AT 20,0: FLASH 1;"SCE
GLI UN NUMERO POI PREMI ENTER"
2755 PLOT 0,0: DRAW 255,0: DRAW
0,175: DRAW -255,0: DRAW 0,-175
2760 INPUT X
2770 CLS : GO TO 2800+10*X
```

La scelta del valore globale è un momento particolarmente delicato dell'esecuzione della procedura di comparazione, infatti da essa dipende il calcolo dei valori percentuali che saranno oggetto della rappresentazione per istogrammi.

Tre sono le alternative e tre sono i sottoprogrammi come del resto è descritto nel diagramma a blocchi discusso in precedenza.

Valore globale somma di dati

La lista delle istruzioni si commenta da sé:

```
2810>REM valore globale somma di
dati
2811 LET q(t+1)=0 FOR i=1 TO t:
```

```

LET q(t+1)=q(t+1)+q(i): NEXT i:
PRINT "valore globale dei dati:
";TAB 8;S$(1,20 TO ),q(t+1): G
O TO 2975
2975>REM calcolo valori percentu
ali

```

In questo caso l'utente dimostra di non volere fare confronto con valori al di fuori di quelli che ha già inserito quindi assume come valore globale la loro somma e limitata la comparazione al confronto diretto dei dati in base alla percentuale con la quale partecipano alla formazione del valore globale.

Inserire valore globale direttamente

L'utente qui è libero di inserire il dato del valore che più lo interessa con la sola limitazione di fornire un valore uguale o superiore alla somma dei valori dei dati già inseriti; in caso contrario all'atto del calcolo dei valori percentuali la loro somma supererà il 100 e il valore globale verrà dichiarato "non accettabile".

Di nuovo la lista delle istruzioni non è da commentare.

```

2820>REM inserisce valore global
e direttamente
2821 LET q(t+1)=0: INPUT "scrivi
valore globale
";q(t+
1): PRINT "valore globale dei da
ti: ";TAB 8;S$(1,20 TO ),q(t+1)
: GO TO 2975
2975>REM calcolo valori percentu
ali

```

Inserire valore globale da Tabella GENERALE

Questa opzione è certo più interessante in quanto consente di utilizzare nuovamente i dati della Tabella GENERALE che in tal caso non solo fornisce i dati da comparare ma anche il termine di paragone.

Una volta scelta una variabile si è prevista una alternativa:

- o il valore globale è la somma dei valori degli stessi dati nella nuova variabile
- o il valore globale è il valore di un nuovo dato.

In entrambi i casi nella tabella devono essere stati inseriti a suo tempo VARIABILI e DATI relativi ad un collettivo di valori; mi spiego: per le tabelle dell'Atlante di lavoro una variabile collettiva va considerata l'Italia e collettivo è il dato reddito lordo regionale e/o nazionale.

Altre variabili e/o dati di questa natura possono essere originati dalle elaborazioni che il programma stesso consente; ad esempio potrebbero essere calcolati i valori dei dati relativi al nord, al centro, al sud Italia oppure per ogni regione le produzioni agricole potrebbero essere accoppiate per prodotti cereali, ortofrutticoli e via di questo passo.

Alla scelta della variabile destinata ad individuare il valore globale si provvede con le seguenti istruzioni

```

2830>REM inserisce valore global
e da tabella generale
2835 INPUT "scrivi la nuova vari
abile per prelevare valore glo
bale":c$(1): PRINT "VARIABILE"
TAB 10; c$(1)
2840 FOR h=1 TO v

```

```

2850 IF c$(1)=b$(h) THEN GO TO 2
875
2860 NEXT h
2870 PRINT "la VARIABILE ";c$(1)
;" non esiste nella tabella GE
NERALE"; PRINT FLASH 1; " PREMI
UN TASTO PER CONTINUARE "; PAUSE
0; GO TO 2730
2875 LET q(t+1)=0: INPUT "il val
ore globale e' la somma del valore
dei dati per la nuova variab
ile? (s/n) ";v$: IF v$="s" THEN
GO TO 2935

```

dopo di che in base alla risposta data all'alternativa posta dall'istruzione 2875 si passa:
 — ad inserire un nuovo dato da accoppiare alla variabile oppure
 — ad eseguire la somma del valore dei dati già inseriti in corrispondenza della variabile scelta.

Il primo caso è risolto dal sottoprogramma:

```

2880>INPUT "scrivi il nuovo dato
per prelevare valore globale"e
$(1); PRINT "DATO";TAB 10; e$(
1); REM cerca nuovo DATO
2890 FOR j=1 TO d
2900 IF e$(1, TO 19)=a$(j, TO 19
) THEN GO TO 2930
2910 NEXT j
2920 PRINT "il DATO ";e$(1, TO 1
9); " non esiste nella tabella GE
NERALE"; PRINT FLASH 1; " PREM
I UN TASTO PER CONTINUARE "; PAU
SE 0; GO TO 2730
2930 LET q(t+1)=a(j,h): PRINT "v
alore globale dei dati: ";TAB 8;
s$(1,20 TO ),q(t+1): GO TO 2975

2975>REM calcolo valori percentu
ali

```

dove è prevista anche la possibilità che il dato prescelto non si trovi nella Tabella GENERALE.

Anche il secondo caso non si presenta di difficile soluzione in quanto i nomi dei dati sono ancora tutti conservati nella memoria bidimensionale e\$ dalla quale è agevole prelevarli per fare la ricerca nella memoria a\$ di Tabella GENERALE. Una volta ottenuto l'incrocio sono noti gli indici poiché presenti nelle memorie j ed h sulla base della procedura di ricerca effettuate, quindi non resta che sommare l'uno dopo l'altro i valori prelevati dalla memoria numerica bidimensionale e comporre in tal modo il valore globale.

Questa è la lista delle istruzioni:

```

2935>REM calcolo somma valori DA
TI per nuova variabile
2940 FOR l=1 TO g
2945 FOR j=1 TO d
2950 IF e$(l, TO 19)=a$(j, TO 19
) THEN LET q(t+1)=q(t+1)+a(j,h)
2955 NEXT j
2960 NEXT l
2965 PRINT "valore globale :";
TAB 8;s$(1,20 TO ),q(t+1)
2975 REM calcolo valori percentu
ali

```

Calcolo valori percentuali

Il sottoprogramma si conclude con questo calcolo che è comune a tutti e tre le opzioni offerte dal menù di scelta del valore globale dei dati:


```

2975>REM calcolo valori percentu
ali
2980 FOR i=1 TO t: LET u(i)=INT
(q(i)/q(t+1)*10000)/100: NEXT i
2985 LET h=0: FOR i=1 TO t: LET
h=h+u(i): NEXT i: IF h>100 THEN
PRINT "valore globale non acce
ttabile": "inserire nuovo valore
": PRINT FLASH 1: PREMI UN T
ASTO PER PROSEGUIRE ": PAUSE 0:
GO TO 2730
2990 RETURN

```

Disegnare istogramma

Questo sottoprogramma si occupa specificatamente della rappresentazione grafica ed a questo scopo provvede in due tempi: prima disegna il sistema di riferimento poi preleva dalla tabella dell'istogramma (indirizzo di memoria u ()) i valori necessari a determinare l'altezza degli istogrammi e li disegna.

La scelta del sistema di riferimento è opzionale tra una scala di valori percentuali con valore massimo 40% ed una con valore 90%; usare l'una o l'altra dipende dal valore percentuale massimo che è presente nella tabella dell'istogramma: alla scelta provvede automaticamente il computer come è facile rilevare dalle prime linee del sottoprogramma:

```

3000>REM DISEGNARE ISTOGRAMMA
3010 PRINT AT 2,2:W$(2,1 TO 10):
PRINT AT 5,1:"val. globale:"AT
7,1:$(1,20 TO 22):"":q(t+1):
PRINT AT 9,2:"%"
3020 FOR i=1 TO t: PRINT AT i,15
i:":$(i,1 TO 7):"":$(i,12
TO 15): NEXT i
3030 PLOT 0,110: DRAW 255,0: PLO
T 115,110: DRAW 0,65: PLOT 0,140
: DRAW 115,0
3040>PLOT 0,0: DRAW 255,0: DRAW
0,175: DRAW -255,0: DRAW 0,-175
3100 PLOT 25,20: DRAW 200,0: PLO
T 25,20: DRAW 0,65: REM assi coo
rdinati
3110 FOR i=1 TO t: REM sceglie s
cala 40% oppure 90%
3112 IF u(i)>40 THEN GO TO 3157
3114 NEXT i
3115>REM disegna in scala 40%
3157>REM disegna in scala 90%

```

Le istruzioni che realizzano gli istogrammi sono simili per entrambe le scale quindi le presentiamo unite:

```

3115>REM disegna in scala 40%
3120 FOR i=1 TO 4: PRINT AT 19-2
+i,1:10*i: PLOT 25,21+i*16: DRAW
200,0: NEXT i: REM scrive scala
di valori % e numeri di identif
icazione
3125 REM disegna istogrammi
3130 FOR i=1 TO t: PRINT AT 20,2
+3*i:
3140 FOR h=1 TO u(i)/10*16: PLOT
15+24+i,20+h: DRAW 8,0: NEXT h
3150 NEXT i
3155 GO TO 3200
3157 REM disegna in scala 90%
3160 FOR i=1 TO 9: PRINT AT 19-i
1,10*i: PLOT 25,21+i*8: DRAW 20
0,0: NEXT i: REM scrive scala di
valori % e numeri di identifica
zione

```

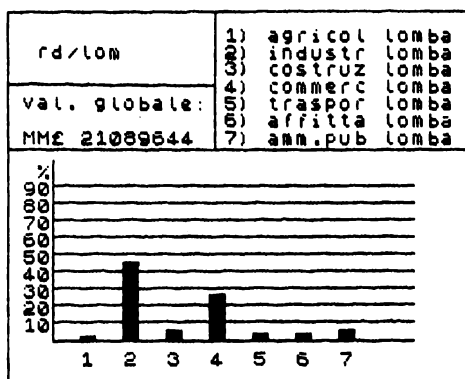
```

3165 REM disegna istogrammi:
3170 FOR i=1 TO t: PRINT AT 20,2
+3*i;
3180 FOR h=1 TO u(i)/10*8: PLOT
16+24*i,20+h: DRAW 8,0: NEXT h
3190 NEXT i
3200 INPUT "stampi l'istogramma?
$ /n ";v$
3210 IF v$="s" THEN COPY
3220 RETURN

```

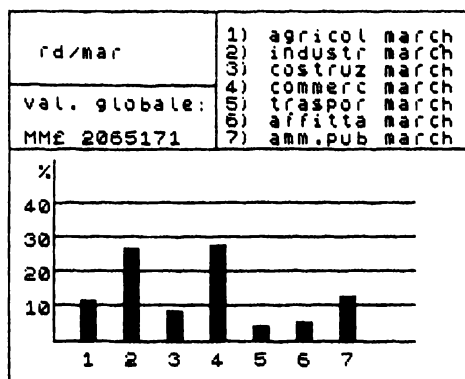
Esaminiamo la linea 3140 (cfr anche 3180) si rileva che la distanza tra due istogrammi adiacenti è di otto pixel come pure è di otto pixel la larghezza dell'istogramma.

Per esemplificare consideriamo gli istogrammi per la formazione del reddito in Lombardia e nelle Marche che abbiamo discusso in apertura di capitolo; ammesso di disporre dei dati in Tabella GENERALE e di averli trasferiti prima per la Lombardia, poi per le Marche nella tabella dell'istogramma, i risultati grafici sono rispettivamente:



valore complessivo dei dati:
MM£ 21089544

1) agricoltur lombardi 3.59 %
2) industria lombardi 45.8 %
3) costruzion lombardi 6.6 %
4) commercio lombardi 28 %
5) trasporti lombardi 4.69 %
6) affittanze lombardi 4 %
7) amm.pubbli lombardi 6.3 %



valore complessivo dei dati:
MM£ 2065171

1) agricoltur marche 11.39 %
2) industria marche 27.19 %
3) costruzion marche 8.8 %
4) commercio marche 28.09 %
5) trasporti marche 4.69 %
6) affittanze marche 5.6 %
7) amm.pubbli marche 13.29 %

A fianco abbiamo riportato i valori percentuali che il programma mette a disposizione con il sottoprogramma LISTARE DATI.

Come si nota, il fatto che in Lombardia la percentuale relativa all'industria supera il 40% fa compiere al computer la scelta di rappresentare sulla scala del 90% piuttosto che su quella del 40%.

Listare dati

I dati sono conservati nella tabella dell'istogramma e nella Tabella GENERALE, il sottoprogramma che esegue la lista di nomi e valori accede ad entrambe le tabelle, come risulta dal menù secondario:

```
LISTARE DATI

per listare dati da tabella
dell'istogramma premi -----> 1

per listare dati da tabella
GENERALE premi -----> 2

per tornare al MENU premi -> 0

SCEGLI UN NUMERO POI PREMI ENTER
```

ottenuto con le seguenti istruzioni:

```
4000>REM LISTARE DATI
4010 PRINT TAB 10;"LISTARE DATI"
4020 PRINT AT 6,1;"per listare d
ati da tabella dell'istogram
ma premi -----> 1"
4030 PRINT AT 10,1;"per listare
dati da tabella GENERALE pre
mi -----> 2"
4040 PRINT AT 15,1;"per tornare
al MENU premi -> 0"
4050 PRINT AT 20,0; FLASH 1;"SCE
GLI UN NUMERO POI PREMI ENTER"
4060 PLOT 0,0: DRAW 255,0: DRAW
0,175: DRAW -255,0: DRAW 0,-175
4080 INPUT n: CLS: IF n=0 THEN
GO TO 5
4090 GO TO 4000+100*n
```

Listare dati da tabella dell'istogramma

Il relativo sottoprogramma non presenta alcuna particolarità: si tratta di far leggere e stampare su video e, se necessario su stampante, il contenuto delle memorie:

- s\$ per il nome dei dati,
- u() per i valori percentuali,
- q() per i valori assoluti,
- w\$(2) per il nome dell'istogramma.

```
4100>REM listare dati da tabella
istogramma
4110 PRINT TAB 3;w$(2)
4120 FOR i=1 TO t: PRINT i;" ";
s$(i, TO 19);TAB 8;s$(i,20 TO )
q(i);NEXT i
4130 PRINT "valore complessiv
o dei dati:";TAB 8;s$(1,20 T
O );q(1+1)
4140 FOR i=1 TO t: PRINT i;" ";
s$(i, TO 19);TAB 8;u(i);"%": NE
XT i
4150 INPUT "vuoi stampare i dati
? (s/n) ";v$: IF v$<>"s" THE
N RETURN
```

```

4155 LPRINT TAB 3;w$(2)"/
4160 FOR i=1 TO t: LPRINT i;" "
;$(i, TO 19);TAB 8;$(i,20 TO
;$(i); NEXT i
4165 LPRINT "/";"valore complessi
vo dei dati:";TAB 8;$(1,20
TO );$(t+1);
4170 FOR i=1 TO t: LPRINT i;" "
;$(i, TO 19);TAB 8;$(i);": N
EXT i
4180 RETURN

```

Nel paragrafo precedente ci siamo già serviti di questo sottoprogramma per visualizzare a fianco dell'istogramma i relativi valori percentuali, nomi dei dati compresi; abbiamo però ommesso l'elenco dei valori assoluti.

Ammessi di avere ancora in memoria la tabella dell'istogramma rd/lom oppure in caso contrario di averlo richiamato da cartridge, l'esecuzione del sottoprogramma fa comparire sul video l'elenco dei dati con i relativi valori.

Listare dati da Tabella GENERALE

In questo caso la soluzione del programma deve essere articolata in quanto la lettura dei dati può essere fatta in più modi: il menù secondario offre all'utente quelli più significativi:

```

LISTARE DALLA TABELLA GENERALE

per listare DATI e valori di
una VARIABILE premi -----> 1

per listare i valori di un
DATO per tutte le VARIABILI
premi -----> 2

per listare i nomi delle
VARIABILI premi -----> 3

per listare i nomi dei
DATI premi -----> 4

per listare la tabella
GENERALE premi -----> 5

per tornare al MENU premi -> 0

SCEGLI UN NUMERO POI PREMI ENTER

```

Al menù provvedono le istruzioni:

```

4200>REM listare dati della tabe
lla generale
4210 PRINT "LISTARE DALLA TABELL
A GENERALE"
4220 PRINT AT 2,1;"per listare D
ATI e valori di una VARIABILE
premi -----> 1"
4230 PRINT AT 5,1;"per listare i
valori di un DATO per tutt
e le VARIABILI premi -----
-----> 2"
4240 PRINT AT 9,1;"per listare i
nomi delle VARIABILI pre
mi -----> 3"
4250 PRINT AT 12,1;"per listare
i nomi dei DATI premi -
-----> 4"
4260 PRINT AT 15,1;"per listare
la tabella GENERALE pre
mi -----> 5"
4265 PRINT AT 18,1;"per tornare
al MENU premi -> 0"

```

```

4270 PRINT AT 20,0: FLASH 1:"SCE
GLI UN NUMERO DEI PREMI ENTER"
4280 PLOT 0,0: DRAW 255,0: DRAW
0,175: DRAW -255,0: DRAW 0,-175
4285 INPUT n:CLS: IF n=0 THEN
GO TO 5
4290 GO TO 4200+100*n

```

Per ciascuna delle opzioni eviteremo di commentare la corrispondente lista delle istruzioni, che del resto non presenta alcuna soluzione particolarmente elaborata, ma presentiamo a fianco delle istruzioni il prodotto, cioè quanto compare sul video.

Nella Tabella GENERALE abbiamo inserito i dati socio-economici delle regioni italiane, quindi sono questi gli elementi che risulteranno dagli elenchi.

Listare dati per una VARIABILE

```

4300>REM listare dati per una va
riabile
4310 DIM c$(1,15): INPUT "scrivi
il nome della variabile ";c$(1
)
4320 FOR i=1 TO v
4330 IF c$(i)=b$(i) THEN GO TO 4
350
4340 NEXT i: PRINT "la VARIABILE
:"TAB 10;c$(1)"non esiste in t
abella GENERALE": RETURN
4350 PRINT "variabile ":"b$(1)"/
: FOR h=1 TO d: PRINT h;"a
$(h,1) TO 19) TAB 8;a$(h,20 TO ):"
:a$(h,1): NEXT h
4360 INPUT "vuoi stampare la lis
ta (s/n) ":"v$: IF v$<>"s" THE
N RETURN
4370 LPRINT "variabile ":"b$(1)"/
: FOR h=1 TO d: LPRINT h;"a
$(h,1) TO 19) TAB 8;a$(h,20 TO ):"
:a$(h,1): NEXT h
4380 RETURN

```

Scelta per VARIABILE la lombardia sul video si ha:

```

variabile : lombardia
1) grano          9203000
2) mais           11373500
3) patate         1040500
4) barbabietole   3500400
5) pomodori       454700
6) mele           419500
7) pesche         105300
8) agrumi         0
9) agricoltura    759287.10
10) industria    9827774.1
scroll?

```

Listare valori di un dato per tutte le VARIABILI

```

4400>REM listare valori di un da
to per tutte le variabili

```

```

4410 DIM a$(1,25): INPUT "scrivi
il nome del dato " :a$(1
)
4420 FOR i=1 TO d
4430 IF a$(1, TO 19)=a$(i, TO 19
) THEN GO TO 4450
4440 NEXT i: PRINT "il DATO : " :T
AB 10;a$(1) : "non esiste in tabel
la GENERALE": RETURN
4450 PRINT "dato : " :a$(i, TO 19)
: FOR h=1 TO v: PRINT h: " ) :b
$(h) :TAB 8;a$(i,20 TO ) : " :a(i,
h) : NEXT h
4460 INPUT "vuoi stampare la lis
ta (s/n) " :v$: IF v$<>"s" THE
N RETURN
4470 LPRINT "dato : " :a$(i, TO 19
) : FOR h=1 TO v: LPRINT h: " )
: b$(h) :TAB 8;a$(i,20 TO ) : " :a(
i,h) : NEXT h
4480 RETURN

```

il DATO scelto è il reddito lordo regionale:

```

dato :reddito lordo
1) valle d'aosta
MM2 259110
2) piemonte
MM2 10144101
3) liguria
MM2 4435165
4) lombardia
MM2 21009644
5) trentino
MM2 1643377
6) veneto
MM2 7100430
7) friuli
MM2 2519721
8) emilia-romagna
MM2 7029972
9) toscana
MM2 6595397
10) marche
MM2 2065171
scroll?

```

Listare nomi VARIABILI

```

4500>REM lista nomi variabili
4510 PRINT TAB 13;"VARIABILI"
: FOR i=1 TO v: PRINT i: " ) :b$(i
) : NEXT i
4520 INPUT "vuoi stampare la lis
ta (s/n) " :v$: IF v$<>"s" THE
N RETURN
4530 LPRINT TAB 13;"VARIABILI"
: FOR i=1 TO v: LPRINT i: " ) :b$
(i) : NEXT i
4540 RETURN

```

```

VARIABILI
1) valle d'aosta
2) piemonte
3) liguria
4) lombardia
5) trentino
6) veneto
7) friuli
8) emilia-romagna
9) toscana
10) marche
11) umbria
12) lazio
13) abruzzo
14) molise
15) campania
16) basilicata
17) calabria
18) sicilia
19) sardegna
20) italia
scroll?

```

Listare nomi DATI:

```

4600>REM lista nomi dati
4610 PRINT TAB 15;"DATI": FOR
i=1 TO d PRINT i;" "a$(i): NE
XT i
4620 INPUT "vuoi stampare la lis
ta (s/n) "v$: IF v$<>"s" THE
N RETURN
4630 LPRINT TAB 15;"DATI": FOR
i=1 TO d LPRINT i;" "a$(i)
NEXT i
4640 RETURN

```

da cui:

DATI		
1)	grano	1700
2)	mais	2000
3)	patate	74300
4)	barbabietole	0
5)	pomodori	0
6)	mele	52400
7)	pesche	0
8)	agrumi	0
9)	agricoltura	MM
10)	industria	MM
11)	costruzioni	MM
12)	commercio	MM
13)	trasporti	MM
14)	affittanze	MM
15)	amm.Pubblica	MM
16)	pianura	KM
17)	reddito lordo	MM
18)	superficie	KM
19)	collina	KM
20)	montagna	KM

scroll?

Lista Tabella GENERALE

```

4700>REM lista tabella GENERALE
4710 PRINT TAB 3;y$(3)
4720 FOR i=1 TO v: PRINT TAB 8,i
a$(i)
4730 FOR h=1 TO d: PRINT h;" "
a$(h,1) TO 19: TAB 8:a$(h,20 TO
a$(h,i): NEXT h
4740 NEXT i
4750 INPUT "vuoi stampare la lis
ta (s/n) "v$: IF v$<>"s" THE
N RETURN
4760 LPRINT TAB 3;y$(3)
4770 FOR i=1 TO v: LPRINT TAB 8,i
a$(i)
4780 FOR h=1 TO d: LPRINT h;" "
a$(h,1) TO 19: TAB 8:a$(h,20 TO
a$(h,i): NEXT h
4785 NEXT i
4790 RETURN

```

da cui:

IT-economica		
1) valle d'aosta		
1)	grano	q 1700
2)	mais	q 2000
3)	patate	q 74300
4)	barbabietole	q 0
5)	pomodori	q 0
6)	mele	q 52400

scroll?

Il risultato in questo caso non è diverso da quello del listare DATI per una VARIABILE, unica differenza è che ora vengono stampati i DATI per tutte le VARIABILI e non per una sola.

Per conservare

Questo sottoprogramma si incarica di trasferire su cartridge il contenuto delle memorie, svolge quindi la funzione opposta a quella del sottoprogramma RICHIAMARE DATI. Il menù:

```

CONSERVARE

per conservare la tabella
dell'istogramma premi -----> 1

per conservare la tabella
GENERALE premi -----> 2

per cancellare file premi -> 3

per tornare al MENU premi -> 0

SCEGLI UN NUMERO POI PREMI ENTER

```

è ottenuto con le istruzioni:

```

5000 REM CONSERVARE
5010 PRINT TAB(10) "CONSERVARE"
5020 PRINT AT 4,1, "per conservare
e la tabella dell'istogramma
premi -----> 1"
5030 PRINT AT 8,1, "per conservare
e la tabella GENERALE premi
-----> 2"
5035 PRINT AT 12,1, "per cancellare
file premi -> 3"
5040 PRINT AT 16,1, "per tornare
al MENU premi -> 0"
5050 PRINT AT 20,0: FLASH 1, "SCE
GLI UN NUMERO POI PREMI ENTER"
5060 PLOT 0,0: DRAW 255,0: DRAW
0,175: DRAW -255,0: DRAW 0,-175
5080 INPUT n: CLS: IF n=0 THEN
GO TO 5
5090 GO TO 5000+100*n

```

I due sottoprogrammi di conservazione delle tabelle rispettivamente dell'istogramma e GENERALE sono in tutto simili a quelli scritti per RICHIAMARE le tabelle dal cartridge; unica differenza sta nel fatto che ora è automatica l'assegnazione del nome del file infatti:

- per la tabella dell'istogramma il nome è composto dalle prime sei lettere del nome completo dell'istogramma conservato nella memoria w\$(2) e vi si aggiungono le desinenze dt1, dt2, dt3, dt4 rispettivamente per le quattro memorie,
- per la Tabella GENERALE il nome si avvale delle prime cinque lettere che compongono il nome della tabella e delle desinenze Gdt1, Gdt2, Gdt3, Gdt4.

Il due sottoprogrammi sono così composti:


```

5100>REM inserire la tabella del
listogramma
5110 INPUT "hai il cartridge nel
drive? s/n" v$
5120 IF v$<>"s" THEN GO TO 5110
5130 INPUT "scrivi il numero del
drive" x$
5140 SAVE "m"x,w$(2, TO 5)+"dt
1" DATA s$(1)
5150 SAVE "m"x,w$(2, TO 5)+"dt
2" DATA q(1)
5160 SAVE "m"x,w$(2, TO 5)+"dt
3" DATA u(1)
5170 SAVE "m"x,w$(2, TO 5)+"dt
4" DATA w$(1)
5180 RETURN
5200 REM conservare tabella gene
rale
5210 INPUT "hai il cartridge nel
drive? s/n" v$
5220 IF v$<>"s" THEN GO TO 5210
5230 INPUT "scrivi il numero del
drive" x$
5240 SAVE "m"x,y$(3, TO 5)+"Gd
1" DATA b$(1)
5250 SAVE "m"x,y$(3, TO 5)+"Gd
2" DATA a$(1)
5260 SAVE "m"x,y$(3, TO 5)+"Gd
3" DATA a(1)
5270 SAVE "m"x,y$(3, TO 5)+"Gd
4" DATA y$(1)
5280 RETURN

```

La terza opzione del menù si spiega in quanto il cartridge non consente di registrare con lo stesso nome due file differenti.

Nasce quindi l'esigenza di cancellare i file già registrati per far posto a quelli aggiornati che vogliamo inserire.

Questo caso si presenta sempre per gli aggiornamenti della Tabella GENERALE che una volta richiamata nel computer ed arricchita con nuovi DATI deve essere reinserita nel cartridge non essendo mutato il nome della tabella e tenuto conto dell'automatismo con cui il programma assegna il nome alla Tabella GENERALE, se quella presente nel cartridge non è stata prima cancellata il microdrive si arresterà segnalando errore per: writing to a read file

Il listato di questo sottoprogramma è elementare quindi non ci soffermiamo a commentarlo:

```

5300>REM cancellare file di dati
5310 INPUT "hai il cartridge nel
drive? s/n" v$
5320 IF v$<>"s" THEN GO TO 5310
5330 INPUT "scrivi il numero del
drive" x$
5340 INPUT "ci sono file da can-
cellare? (s/n)" v$
5350 IF v$<>"s" THEN CLS : GO TO
5000
5360 INPUT "scrivi il nome del f
ile che vuoi cancellare tralascia
ndo dt1"
5370 PRINT v$, "va bene? s/n"
5380 IF INKEY$<>"s" THEN GO TO 53
60
5385 IF INKEY$="" THEN GO TO 538
0
5390 IF INKEY$<>"s" THEN GO TO 5
360
5400 ERASE "m"x,v$+"dt1"
5410 ERASE "m"x,v$+"dt2"
5420 ERASE "m"x,v$+"dt3"
5430 ERASE "m"x,v$+"dt4"
5440 RETURN

```

Per correggere

L'utilità di un sottoprogramma di questo tipo è limitata alla correzione degli errori di battitura nei nomi delle **VARIABILI** o dei **DATI** in Tabella **GENERALE**, infatti se si dovesse modificare il valore di un **DATO** in corrispondenza di una **VARIABILE** si può far uso del sottoprogramma

INSERIRE DATI IN TABELLA GENERALE

Quest'ultimo, quando inserisce i dati nella tabella, se trova la posizione indicata dall'utente occupata da un altro valore, lo sostituisce con quello appena ricevuto.

Data la limitata funzione e la semplicità di scrivere una lista di istruzioni adatta allo scopo ci permettiamo di omettere la soluzione e lasciarla al lettore per esercizio.

Percorsi didattici

4.1 Un esempio di simulazione: la statistica dei grandi numeri spiegata con carta e matita e con il computer

SERGIO VETTORE

4.1.1 Obiettivi didattici

I programmi della scuola media prevedono in una delle loro prime stesure la necessità di affrontare a scuola argomenti tali da portare gli alunni alla comprensione “della differenza fra il certo e il probabile, il continuo e il discontinuo, leggi matematiche e leggi empiriche”.

Nella stesura definitiva degli stessi compare fra i temi da sviluppare nel triennio per scienze matematiche, chimiche, fisiche e naturali quello degli “avvenimenti casuali, nozioni di probabilità e sue applicazioni”, tema ormai presente in tutti i libri di testo di matematica data l'enorme rilevanza che “probabilità e statistica” hanno assunto nel campo matematico, scientifico, economico, sociale, ecc.

Tema oltremodo importante per chi vuole lavorare in classe anche con l'elaboratore.

Affrontando questo argomento e avendo a disposizione in classe uno o più elaboratori si possono far raggiungere o consolidare agli alunni i seguenti obiettivi:

- a) saper ideare e progettare una sperimentazione,
- b) saper registrare, ordinare, correlare ed elaborare dati
- c) saper codificare e decodificare i vari codici della comunicazione,
- d) saper collegare logicamente i fatti e i dati raccolti da diverse fonti e sperimentazioni, al fine di formulare ipotesi più complete
- e) saper verificare se vi è corrispondenza fra ipotesi formulate e risultati sperimentali,
- f) saper inquadrare in un medesimo schema logico questioni diverse.

Spesso nell'affrontare in classe avvenimenti casuali e nozioni di probabilità si utilizza il lancio di monete e dadi, urne contenenti palline o altro materiale.

È utile partire da esperienze di questo tipo: possiamo così organizzare attività di laboratorio anche quando si affrontano argomenti matematici.

Durante alcuni di questi esperimenti i ragazzi saranno guidati a:

- a) raccogliere sistematicamente dati,
- b) farne rappresentazioni grafiche,
- c) individuare l'insieme dei casi possibili,
- d) confrontare i propri dati con i dati provenienti da altri o da libri e riviste ecc.
- e) determinare la probabilità matematica,
- f) determinare la probabilità sperimentale o frequenza relativa,
- g) determinare lo scarto,
- f) intuire la legge dei grandi numeri.

Tutti i punti indicati sono certamente raggiungibili attraverso un lavoro fatto in classe con “carta e matita”, non vi è necessità dell’elaboratore.

Che fare, però, quando si debbono raccogliere, tabulare, analizzare quantità enormi di dati? Quando si devono fare migliaia di lanci, ad esempio, di due monete?

Oppure nel caso del lancio di 2 dadi, come celermente calcolare probabilità sperimentale, scarto relativo a migliaia e migliaia di lanci? Come verificare la legge dei grandi numeri?

Diviene a questo punto utile l’utilizzo dell’elaboratore, elaboratore che qualcuno ha ad esempio istruito a:

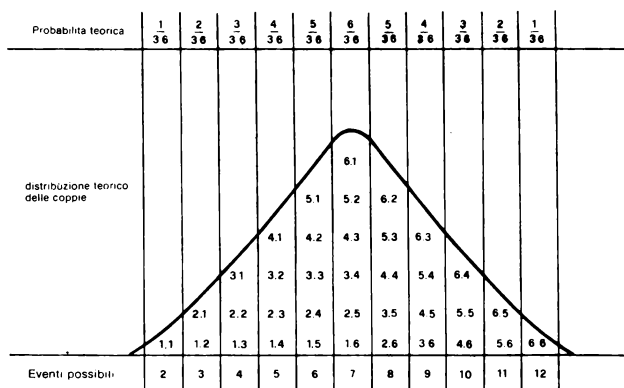
- simulare “n” lanci di monete o di dadi,
- individuare i casi possibili,
- memorizzare le varie frequenze,
- incorporare nelle proprie memorie le varie previsioni matematiche,
- calcolare probabilità sperimentale e scarto,
- tracciare istogrammi.

In definitiva si tratta di utilizzare (e insegnare agli alunni ad utilizzare) in classe un programma già preconfezionato (scritto dal docente) relativo alla simulazione del lancio di monete o dadi (i cui listati sono riportati nella parte conclusiva di questo capitolo).

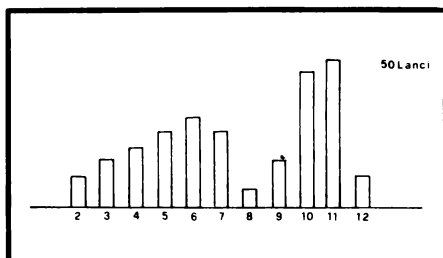
È utile anche questo approccio con l’elaboratore? Certamente, basta considerare solo *alcuni* esempi:

- il computer diviene uno degli strumenti “confidenzialmente” utilizzato all’alunno,
- l’alunno si trova di fronte ad un’altra fonte di dati (veloce, potente, ripetibile, flessibile, ecc.) da leggere, interpretare, confrontare, ecc.
- vengono rafforzate le capacità operative degli alunni nei confronti dell’elaboratore: caricare un programma su nastro, riportarlo nelle memorie dell’elaboratore, rispondere a precisi comandi, ecc.
- l’alunno dopo aver praticamente sperimentato il lancio di due dadi, posto davanti al video di un elaboratore istruito alla simulazione del lancio di 2 dadi e liberato dalla “fatica” (sempre necessaria) di alcune operazioni pratiche e ripetitive, può meglio concentrarsi su analisi di dati e risultati, su procedure di confronto, ecc. Un esempio:

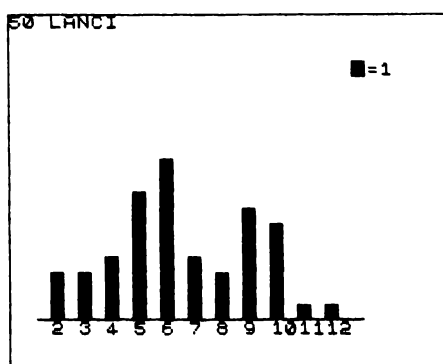
Questo è l’istogramma “distribuzione teorica coppie”, relativo al lancio di due dadi, tracciato dagli alunni:



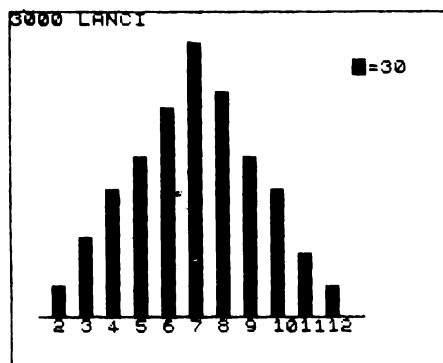
Questo è l'istogramma tracciato da alunni relativo a 50 lanci sempre di due dati:



Questo è l'istogramma tracciato dall'elaboratore relativo a 50 lanci di 2 dadi:



Questo è l'istogramma tracciato dall'elaboratore relativo a 3000 lanci:



Dall'analisi comparata dei 4 istogrammi (2 prodotti dall'alunno, 2 prodotti dall'elaboratore) gli alunni possono essere immediatamente guidati a comprendere che:

a) esiste una differenza fra distribuzione teorica e distribuzione sperimentale degli eventi,

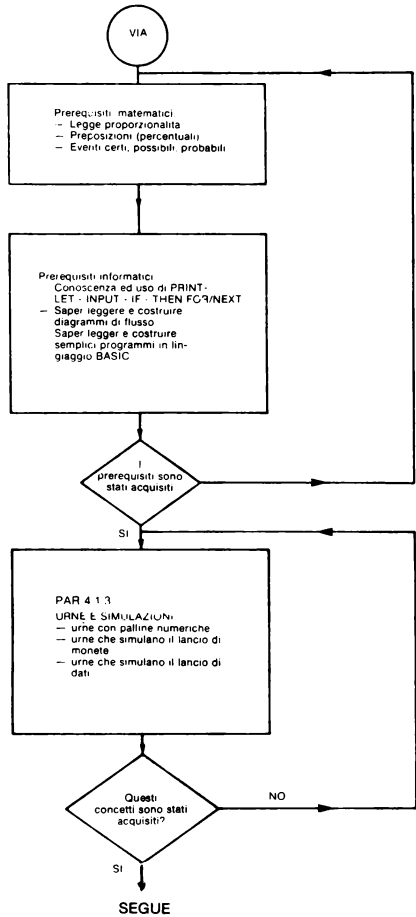
- b) l'evento sperimentale non riproduce *quasi mai* la distribuzione teorica,
- c) all'aumentare del numero di lanci la frequenza di un certo evento si avvicina sempre più al valore della probabilità teorica.

4.1.2 Itinerario didattico

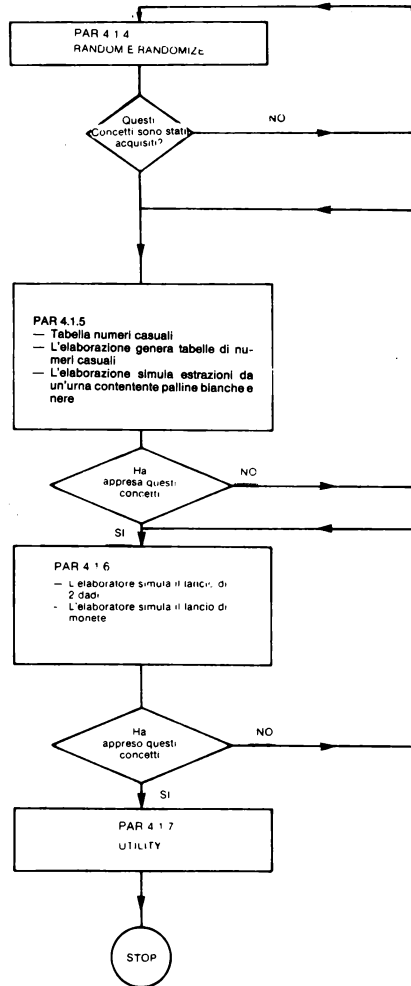
Quello indicato nel precedente paragrafo è uno dei modi in cui si può utilizzare l'elaboratore in classe, ma se resta l'unico non portiamo gli alunni ad una maggiore comprensione del funzionamento dell'elaboratore, della sua programmabilità, ecc.

Come fa un elaboratore a simulare il lancio di monete o dadi?

Seguendo l'itinerario sotto riportato forse riusciremo a rispondere alla domanda, postaci concretamente da un alunno, e a fargli raggiungere alcuni degli obiettivi indicati.



SEGUITO



4.1.3 Urne e simulazioni

Con questo capitolo cercheremo di far comprendere ai ragazzi, lavorando su urne di diverso tipo, come e perché un evento incerto, di determinata probabilità, possa essere simulato (sostituito) da altro evento incerto avente medesima probabilità.

Consideriamo quest'urna contenente 7 palline bianche e 3 nere:

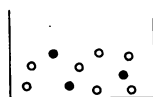


domandiamo ai ragazzi:

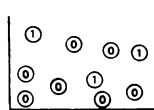
- qual è la probabilità teorica di estrarre una pallina nera? ($3/10$).
- qual è la probabilità teorica di estrarre una pallina bianca? ($7/10$).

Possiamo poi far eseguire un certo numero di estrazioni: determinare la frequenza, calcolare le probabilità sperimentali e gli scarti.

Non sempre abbiamo a disposizione palline bianche e palline nere, cerchiamo qualcosa di più comodo e pratico.



URNA A



URNA B

L'urna B è formata da 3 dischetti con il numero 1, e 7 dischetti con il numero 0.

Poniamo ai ragazzi i seguenti quesiti:

- a) qual è la probabilità teorica di estrarre una pallina nera dall'urna A? ($3/10$);
- b) qual è la probabilità teorica di estrarre il numero 1 dall'urna B? ($3/10$);
- c) qual è la probabilità teorica di estrarre una pallina bianca dall'urna A? ($7/10$);
- d) qual è la probabilità teorica di estrarre il numero 0 dall'urna B? ($7/10$).

Poniamo ora questa *condizione*: ogni volta che estraiamo dall'urna B il numero *uno* è come se estraessimo una pallina nera dall'urna A (i numeri zero rappresenteranno, ovviamente, le palline bianche).

Immaginiamo di aver eseguito le seguenti 10 estrazioni dall'urna B:
0 0 1 0 1 1 0 0 0 0

esse corrispondono alle seguenti estrazioni dall'urna A:

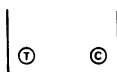
○○●●●○○○

Invece di operare sull'urna A possiamo tranquillamente operare sull'urna B molto più comoda e reperibile.

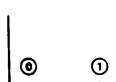
CONCLUSIONE: Pur agendo su urne con oggetti diversi (palline-numeri), generiamo eventi con *medesima probabilità teorica*. Possiamo allora dire che le *estrazioni dall'urna B simulano (sostituiscono) estrazioni dall'urna A*.

Monete e urne 0,1

Consideriamo queste due urne:

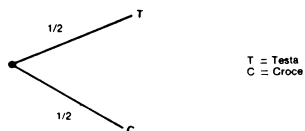


URNA A



URNA B

a) Già sappiamo che lanciando una moneta si hanno: eventi possibili, probabilità teoriche, indicati dal seguente diagramma ad albero:



T = Testa
C = Croce

- b) Il lancio di una moneta può essere “simulato” dall'estrazione di un dischetto dall'urna A.
 c) Il lancio di una moneta può essere “simulato” dall'estrazione di un dischetto numerato dall'urna B, ponendo queste condizioni:

0 → T
 1 → C

NB: Nei casi precedenti e nei successivi l'estrazione dovrà sempre avvenire con le seguenti modalità:

- 1) il contenuto dell'urna non deve essere visibile,
- 2) si estrae un oggetto per volta,
- 3) ogni oggetto, dopo l'estrazione, va ricollocato nell'urna,
- 4) prima di una successiva estrazione l'urna va agitata.

Due monete - urne 0 e 1



- a) Nel caso del lancio di 2 monete i casi possibili e le probabilità teoriche dei vari eventi sono:

		II MONETA	
		T	C
I MONETA	T	T.T	T.C
	C	C.T	C.C

$P(TT) = 1/4 = 25\%$
 $P(CC) = 1/4 = 25\%$
 $P(TC/TC) = 2/4 = 50\%$

- b) Il lancio di due monete può essere simulato dalle urne di fig. a e b estraendo, contemporaneamente, un dischetto dalla 1° e 2° urna.

Consideriamo le urne di fig. b estraendo, da ognuna di esse, un dischetto per volta, avremo i seguenti casi possibili:

		II URNA	
		0	1
I URNA	0	0.0	0.1
	1	1.0	1.1

e le seguenti probabilità teoriche:

$$P(0,0) = 1/4 = 25\%$$

$$P(1,1) = 1/4 = 25\%$$

$$P(0,1/1,0) = 2/4 = 50\%$$

CONCLUSIONE: anche in questo caso con le urne di figura *a* e *b* abbiamo potuto simulare un certo evento: il lancio di 2 monete.

2 Dadi - urne numeriche

a) Ogni volta che lanciamo due dadi e addizioniamo i valori numerici che appaiono, abbiamo i seguenti casi possibili e le seguenti probabilità teoriche:

+	1	2	3	4	5	6
1	2	3	4	5	6	7
2	3	4	5	6	7	8
3	4	5	6	7	8	9
4	5	6	7	8	9	10
5	6	7	8	9	10	11
6	7	8	9	10	11	12

$$P(2) = P(12) = 1/36$$

$$P(3) = P(11) = 2/36$$

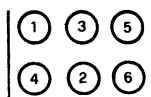
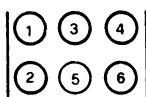
$$P(4) = P(10) = 3/36$$

$$P(5) = P(9) = 4/36$$

$$P(6) = P(8) = 5/36$$

$$P(7) = 6/36$$

b) consideriamo le urne della figura che segue:



Estraendo da ognuno di esse un dischetto numerato e addizionando i valori numerici che verranno estratti, avremo i medesimi casi possibili e le medesime probabilità teoriche riferite al lancio di 2 dadi sopra indicati

CONCLUSIONE: Anche in questo caso, attraverso estrazioni dalle urne numeriche abbiamo “simulato” il lancio di due dati.

NB: È evidente che con tutte le urne che abbiamo precedentemente indicato possiamo proporre agli alunni “simulazioni” di “n” lanci con monete o dadi e i relativi calcoli: frequenza, probabilità sperimentali, scarti.

4.1.4 Urne ed elaboratore (RND)

Incominciamo ora ad utilizzare l’elaboratore cercando di far comprendere agli alunni il significato ed uso della funzione BASIC chiamata RANDOM (RND), funzione che permette “la generazione di numeri casuali”.

Che cosa vuol dire “generare numeri casuali”?

a) si definiscono *casuali* tutti quei *fenomeni* per i quali la risposta *non è prevedibile* da un istante a quello successivo;

b) una *grandezza variabile*, *varia casualmente*, quando in ogni momento assume valori dipendenti solo ed esclusivamente dal caso.

Dopo aver fissato l’attenzione degli alunni sui punti a) e b) (ricollegandoci anche alle esperienze già svolte in classe), mandiamo in esecuzione questo programma in linguaggio BASIC:

```
10 PRINT RND,
20 GO TO 10
```

Premendo RUN + ENTER sul video appariranno numeri di questo tipo:

0.44711304	0.53410339
0.056273315	0.37158577
0.86822251	0.11701965
0.7493746108	0.31132507
0.0000815869	0.26263428
0.000041003	0.38108828
0.000032117	0.67455055
0.0000240723	0.43099976
0.000022256	0.42245483
0.0000476868	0.35800171
0.00008606	0.81471252
0.1036377	0.77384949
0.038955688	0.92277527
0.200822144	0.61750793
0.31352234	0.51495361
0.62007031	0.65570068
0.17783274	0.34586623
0.94221497	0.66618347
0.26414155	0.31066895
0.30093491	0.57197571
0.69666638	0.40008945
0.070800781	0.53213501

Cosa è avvenuto? Nel momento in cui l'istruzione 10 viene eseguita l'elaboratore genera un valore numerico casuale compreso tra zero e uno.

Per avere un'idea di ciò che avviene immaginiamo di avere un'urna contenente un certo numero di palline bianche e di operare con un elaboratore che con la funzione RND generi numeri di 6 cifre e di scrivere con 6 cifre (zero compreso) ogni possibile frazione decimale sulle nostre palline bianche.

Avremo allora costruito un'urna contenente 10^6 palline bianche numerate con frazioni diverse.

Ogni volta che, durante l'esecuzione di un programma, l'elaboratore legge un'istruzione contenente la funzione RND è come se pescasse una pallina a caso dentro questa enorme urna.

Dopo ogni estrazione la pallina viene collocata nell'urna; affinché la probabilità di estrarre una qualsivoglia frazione decimale sia sempre $1/10^6$, l'urna viene scossa per rimescolare le palline.

Con questo semplice programma:

```
10 PRINT RND
20 GO TO 10
```

abbiamo allora determinato l'ipotetica generazione di un'urna contenente numeri maggiori di 0 ma minori dell'unità.

Da quest'urna l'elaboratore estrae un numero, ne stampa il valore sul video, lo ricolloca nell'urna (tutto ciò può essere interrotto solo se premiamo contemporaneamente CAPS-SHIFT e BREAK).

È evidente che l'elaboratore non costruisce urne nè esegue estrazioni, questo è un "modello interpretativo" della funzione RND per far meglio comprendere agli alunni ciò che sta avvenendo.

In realtà mandando, ad esempio, in esecuzione questo programma:

```
10 LET A=RND
20 PRINT A
```

ogni volta che premiamo RUN+ENTER l'elaboratore *genera automaticamente* (per i nostri alunni estrae dall'urna) un valore numerico casuale compreso tra zero e uno.

Questo valore viene posto nella memoria di indirizzo A e (linea 20) stampato sul video. Come è possibile generare numeri casuali? Facciamo due esempi:

Con carta e matita

Dalla geometria sappiamo che:

$\frac{\text{Circonferenza}}{\text{Diametro}} = 3,141592653589793238462.....$

Diametro

andando sempre avanti nella definizione del valore di π otteniamo certamente numeri casuali visto che, per ciò che se ne sa, il numero non presenta una sequenza di cifre prevedibili in base a qualche particolare andamento razionalmente individuabile.

Con elaboratore

Von Neuman fu uno dei primi che progettò un algoritmo capace di determinare la generazione di numeri pseudocasuali (algoritmo traducibile in un programma comprensibile per qualsiasi elaboratore).

L'esempio di algoritmo che consideriamo ha molti limiti, ha quindi solo il compito di far intuire la possibilità di programmare un elaboratore affinché generi numeri casuali (in realtà pseudo-casuali).

Consideriamo il seguente numero di quattro cifre: 5432.

- a) elevandolo al quadrato si otterrà: 29506624, numero composto da 8 cifre;
- b) consideriamo le quattro cifre centrali: 5066;
- c) 5066 è il *primo numero casuale* generato dall'elaboratore. Da esso viene generato il successivo;
- d) elevando al quadrato 5066 si otterrà 25664356;
- e) le quattro cifre centrali (6643) rappresentano il *secondo numero casuale*;
- f) e così via.

Si potrebbe continuare all'infinito, in realtà utilizzando 4 cifre si possono ottenere solo 9999 numeri diversi fra loro, per cui, la sequenza è certamente infinita ma ad un certo punto si ripete. Anche nei moderni elaboratori la sequenza ad un certo punto si ripete, ma ciò avviene in tempi così lunghi (generando numeri in continuazione alcuni mesi) che, per l'uso che noi facciamo dell'elaboratore, è come se la ripetizione non avvenisse mai.

Con l'esempio precedente abbiamo compreso che un elaboratore per generare numeri casuali *esegue un programma*, cioè una *precisa sequenza di operazioni*, complesse fin che si vuole, ma *non imprevedibili*, cioè non perfettamente casuali.

Poiché si ha produzione di numeri casuali solo quando in ogni momento, essi assumono valori dipendenti solo ed esclusivamente dal caso, un elaboratore può produrre solo numeri pseudo-casuali.

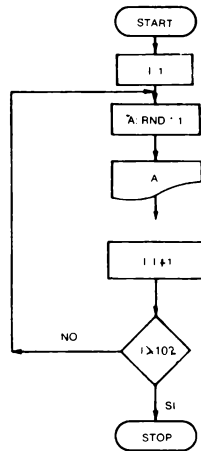
Ed è per questo che nel manuale dello SPECTRUM (Alla scoperta dello ZX SPECTRUM- Gruppo Editoriale Jackson, p. 125) possiamo leggere: "RND non ritorna numeri perfettamente casuali perché segue una sequenza prefissata di 65536 numeri casuali. In ogni caso questi numeri sono mescolati così bene, che non vi sono apparentemente regole di generazione; diciamo quindi che RND è una funzione pseudo-casuale".

Ma è una pseudo-casualità così sofisticata che può tranquillamente sostituire (per i nostri algoritmi) una generazione di numeri veramente casuale.

Ritorniamo ai nostri alunni. Scriveremo un programma con il quale istruiamo il nostro elaboratore ad eseguire solo 10 estrazioni da un'ipotetica urna.

Per ottenere ciò è sufficiente una semplice iterazione (FOR/NEXT).

Il seguente diagramma di flusso chiarisce molto bene come possiamo risolvere questo problema:



Il programma che traduce questo diagramma di flusso è:

```
10 FOR I=1 TO 10
20 LET A=RND*1
30 PRINT A
40 NEXT I
```

Mandando in esecuzione questo programma appariranno sul video 10 valori numerici casuali compresi fra 0 ed 1 escluso.

Anche questa volta abbiamo fatto costruire al nostro elaboratore un'ipotetica urna numerica dalla quale eseguirà soltanto 10 estrazioni (infatti il ciclo FOR/NEXT ci permette di contare il numero di estrazioni che vengono effettuate e di interromperle dopo la decima).

Mandiamo ora in esecuzione questo programma, che differisce dal precedente solo nella linea 20 ove, invece di RND*1, troviamo RND*2.

Avremo quindi:

```
10 FOR I=1 TO 10
20 LET A=RND*2
30 PRINT A
40 NEXT I
```

Abbiamo così istruito il nostro elaboratore a:

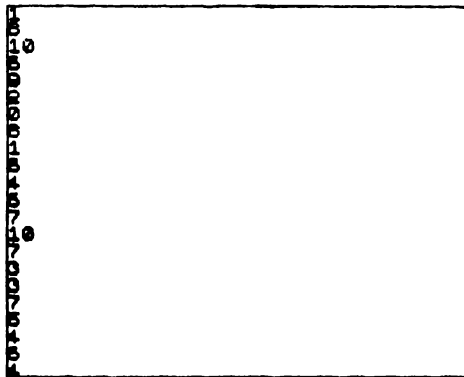
- costruire un'ipotetica urna contenente i numeri compresi fra 0 e 2 escluso;
- eseguire 10 estrazioni;
- stampare sul video i numeri estratti.

Vogliamo ora costruire un'urna *contenente solo numeri interi*.

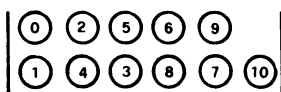
Consideriamo questo programma:

```
10 LET A=INT (RND*11)
20 PRINT A
```

Mandandolo in esecuzione più volte vedremo apparire sullo schermo numeri di questo tipo:



Infatti, attraverso questo programma è come se l'elaboratore, fra tutti i numeri maggiori di zero, ma minori di 11, considerasse, per costruire l'urna, *solo quelli interi*. Le estrazioni avverrebbero allora da un'urna di questo tipo:



La funzione BASIC INT presente alla linea 10 ha appunto il compito di definire *la sola parte intera di un numero*.

In linguaggio informatico il precedente programma ha questo significato:

- a) vengono generati valori numerici casuali compresi fra 0 ed 11 escluso;
- b) attraverso la funzione INT viene posto nella memoria di indirizzo A il solo valore intero.

Il risultato di queste operazioni è la generazione casuale di numeri interi compresi fra 0 e 10.

Finora abbiamo imparato a far costruire ipotetiche urne contenenti numeri interi e/o decimali sempre compresi fra zero e altro numero, maggiore di zero, intero o decimale.

Cerchiamo ora di far costruire urne contenenti valori numerici compresi fra due numeri, interi o decimali, da noi arbitrariamente scelti (in linguaggio informatico: far generare all'elaboratore valori numerici casuali compresi fra due numeri da noi arbitrariamente scelti).

Se con $RND*5$ determiniamo la costruzione di un'ipotetica urna contenente valori numerici compresi fra 0 e 5 escluso (facciamo casualmente generare valori numerici compresi fra 0 e 5 escluso).

Con $(RND*5)+1$ determiniamo la costruzione di un'ipotetica urna contenente valori numerici compresi fra 1 e 6 escluso (facciamo casualmente generare numeri compresi fra 1 e 6 escluso).

Con $(RND*5)+2$ nell'urna vi saranno valori numerici compresi fra 2 e 7 escluso.

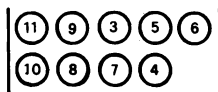
È allora evidente che con questo programma:

```

10 FOR I=1 TO 10
20 LET A=INT (RAND*9) +3
30 PRINT A
40 NEXT I

```

determiniamo l'ipotetica generazione di quest'urna, dalla quale vengono eseguite 10 estrazioni.



Randomize

Come abbiamo già detto, RND produce una sequenza di numeri casuali che è il risultato di una precisa successione di operazioni matematiche dove *ogni numero è ricavato dal precedente*.

Essendo il generatore di numeri casuali un programma, è chiaro che necessita di *dati d'ingresso*, nel caso specifico di un particolare numero (SEME=SEED) dal quale iniziare la sequenza di operazioni matematiche (catene numeriche casuali) che genereranno la sequenza di numeri casuali.

In mancanza di questo seme il generatore di numeri casuali parte da un numero che si trova nelle memorie dell'elaboratore; nello SPECTRUM *questo numero è sempre lo stesso*.

Se vogliamo inizializzare la funzione RND (cioè farlo partire da numeri sempre diversi) si deve usare il comando RANDOMIZE (RAND *da solo* equivalente a RANDOMIZE 0) che va posto prima di RND.

Ai programmi precedenti va quindi aggiunta questa istruzione

5 RANDOMIZE

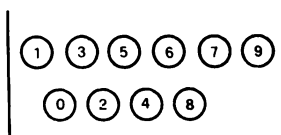
4.1.5 Tabella numeri casuali - urne - elaboratore

Consideriamo una tabella di questo tipo:

12159	66144	05091	13446	45653	13684	66024	91410	51351	22772
30156	90519	95785	47544	66735	35754	11088	67310	19720	08379
59069	01722	53338	41942	65118	71236	01932	70343	25812	62275
54107	58081	82470	59407	13475	95872	16268	78436	39251	64247
99681	81295	06315	28212	45029	57701	96327	85436	33614	29070
27252	37875	53679	01889	35714	63534	63791	76342	47717	73684
93259	74585	11863	78985	03881	46567	93696	93521	54970	37607
84068	43759	75814	32261	12728	09636	22336	76529	01017	45503
68582	97054	28251	63787	57285	18854	45006	16343	51867	67979
60646	11298	19680	10087	66391	70853	24453	73007	74958	29020
97437	52922	80739	59178	59628	61017	51652	40915	94696	67843
58009	20681	98823	50979	01237	70152	13711	73916	87902	84759
77211	70110	93803	60135	22881	13423	30999	07104	27400	25414
52256	84591	65302	99257	92970	28924	36632	54044	91798	78018
37493	69330	94069	39544	14050	03476	25804	49350	92525	87941
87569	22661	55970	52623	35419	76660	42394	62310	62626	00581
22896	62237	39635	63725	10463	87944	92075	90914	30599	35671
02697	33230	64527	97210	41359	79399	13941	88378	68503	33609
20080	15652	37216	00679	02088	34138	13953	68939	05630	27653
20550	95151	60557	57449	77115	87372	02574	07851	22428	39189

72771	11672	67492	42904	64647	94354	45994	42538	54885	15983
38472	43379	76295	69406	96510	16529	83500	28590	49787	29822
24511	56510	72654	13277	45031	42235	96502	25567	23653	36707
01054	06674	58283	82831	97048	42983	06171	12350	49990	04809
94437	94907	95274	26487	60496	78222	43032	04276	70800	17378
97842	69095	25982	03484	25173	05982	14682	31653	17170	92785
53047	13486	69712	33567	82313	87631	03197	02438	12374	40329
40770	47013	63306	48154	80970	87976	04939	21233	20572	31013
52733	66251	69661	58387	72096	21355	51659	19003	75556	33095

Essa è chiamata *tabella dei numeri casuali* e viene spesso utilizzata per il calcolo delle probabilità, infatti simula, molto comodamente, estrazioni dalla seguente urna contenente 10 palline numerate:

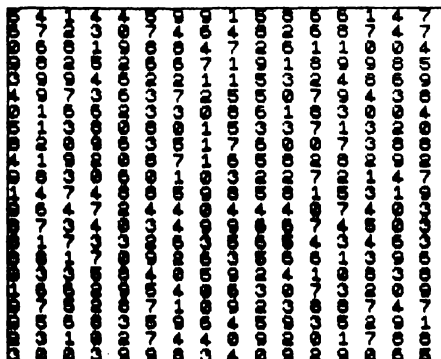


Una tabella simile si può ottenere in classe estraendo da quest'urna una pallina alla volta. Dopo ogni estrazione la pallina viene ricollocata nell'urna, prima di una successiva estrazione le palline devono essere rimescolate affinché la probabilità di estrarre una qualsiasi delle cifre presenti nell'urna sia sempre $1/10$.

Anche con l'elaboratore possiamo ottenere una tabella simulante "n" estrazioni dall'urna. Basta far girare questo programma:

```
10 PRINT INT (RND*10); " ";
20 GO TO 10
```

e sul video apparirà una tabella di numeri casuali del tutto simile a quella precedente.

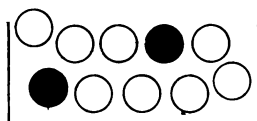


Anche questa volta abbiamo fatto costruire al nostro elaboratore un'ipotetica urna contenente i numeri: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, dalla quale estrae un numero alla volta, il numero viene stampato sul video e ricollocato nell'urna.

Dal punto di vista del linguaggio BASIC con la linea 10 abbiamo istruito l'elaboratore a:

- generare numeri casuali compresi fra 0 e 10 escluso (RND*10);
- prenderne la sola parte intera: INT (RND*10);
- stamparne il valore intero sul video (PRINT).

È possibile utilizzare la tabella dei numeri casuali, prodotti dagli alunni e dall'elaboratore per simulare estrazioni dall'urna sottostante?



Certamente, basta decidere che, nella tabella di pag. 161, i numeri 0 e 1 rappresentano estrazioni di palline nere.

Leggendo nella tabella dei numeri casuali i primi 50 numeri (seguendo le linee orizzontali da sinistra verso destra), è come se eseguiamo le seguenti 50 estrazioni dall'urna delle palline numerate (pag. 162).

12159 66144 05091 13446 45653 13604 60024 91410 51351 22772

Questi numeri simulano quindi le sottostanti 50 estrazioni dall'urna con palline bianche e nere presentata sopra:

●○○○○○●○○●○○●○○○○○○○○○○●○○○●○○○●○
●●○○○○●○○○○○

Proprio comoda questa "tabella dei numeri casuali" non abbiamo più bisogno di urne, di palline, di estrazioni: basta porre alcune condizioni, leggere i numeri e immediatamente possiamo simulare "n" estrazioni da urne contenenti 10 palline di diverso colore!

Ma possiamo fare anche qualcosa di meglio:

- l'urna ci dice che la *probabilità teorica* di estrazione di una pallina nera o bianca è:

$$P(N) = 2/10 = 0,2 = 20\%$$

$$P(B) = 8/10 = 0,8 = 80\%$$

- I 50 numeri letti sulla riga orizzontale della "tabella dei numeri casuali" ci dicono quante palline bianche e quante palline nere sono apparse su 50 estrazioni:

n° palline bianche:	36
n° palline nere:	14
	<hr/> 50

- possiamo allora calcolare la *probabilità sperimentale e scarto*:

probabilità sperimentale	$(N)=14/50=0,28=28\%$
probabilità teorica	$(B)=36/50=0,72=72\%$
scarto (N)	$0,28-0,2=+0,08 \quad (+8\%)$
scarto (B)	$0,72-0,8=-0,08 \quad (-8\%)$

Tutto ciò che abbiamo ricavato nei punti a), b), c) non è stato ottenuto eseguendo delle vere estrazioni, ma simulandole mediante la lettura della tabella dei numeri casuali.

I ragionamenti ora fatti ci permettono anche di comprendere come sarà possibile istruire un elaboratore alla simulazione di estrazioni da urne, al calcolo delle probabilità.

NB: è evidente che la tabella dei numeri casuali può essere letta in modi diversi: linee orizzontali, linee verticali, linee oblique, destra sinistra, sinistra destra, ecc.

Quanto abbiamo fatto nei precedenti paragrafi utilizzando la "tabella dei numeri casuali" è ottenibile attraverso un elaboratore opportunamente istruito?

Consideriamo il seguente programma:

```
10 FOR I=1 TO 50
20 PRINT INT (RND*10);
30 NEXT I
```

mandandolo in esecuzione potrebbe apparire questa serie di 50 numeri naturali (numeri sempre compresi fra 0 e 9):

```
478725817618688558987332489817
7888948417811189813
```

È come se l'elaboratore avesse letto 50 numeri sulla "tabella dei numeri casuali".

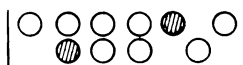
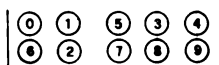
Con il precedente programma determiniamo:

- la generazione di un'ipotetica urna contenente i numeri 0,1,2,3,4,5,6,7,8,9 dalla quale l'elaboratore eseguirà 50 estrazioni;
- la loro stampa sul video

Perfezioniamo il programma precedente ponendo le seguenti condizioni:

- i numeri 0 ed 1 rappresentano estrazioni di palline nere;
- gli altri numeri rappresentano estrazioni di palline bianche.

Ponendo queste due condizioni le effettive estrazioni eseguite dall'ipotetica urna A simulano estrazioni dall'urna B:



Con questo programma:

```
1 RANDOMIZE
10 FOR I=1 TO 50
20 LET A=INT (RND*10)
30 IF A=0 OR A=1 THEN PRINT "■"
40 IF A<>0 AND A<>1 THEN PRINT "0"
50 NEXT I
```

diciamo all'elaboratore:

- a) crea un'ipotetica *urna A* dalla quale eseguirai 50 estrazioni. Dopo ogni estrazione i numeri estratti vengono ricollocati nell'urna.
- b) se dall'urna A etrai il numero zero o il numero 1 stampa sul video "■" (è come aver estratto una pallina nera dall'urna B);
- c) se dall'urna B estrai numeri diversi da zero e da uno stampa sul video " 0 " (è come aver estratto una pallina bianca dall'urna B);

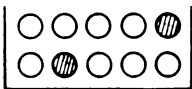
Analizziamo questo programma dal punto di vista del linguaggio informatico:

linea 10: ad ogni ciclo FOR/NEXT l'elaboratore genera un valore numerico casuale compreso fra zero e nove.

linea 20: Questo valore numerico viene posto nella memoria di indirizzo A. se nella memoria di indirizzo A è contenuto il valore numerico uno o il valore numerico zero, stampante sul video: "■".

linea 30: se nella memoria di indirizzo A è contenuto un valore numerico diverso da zero e diverso da uno, stampare: "0".

CONCLUSIONE

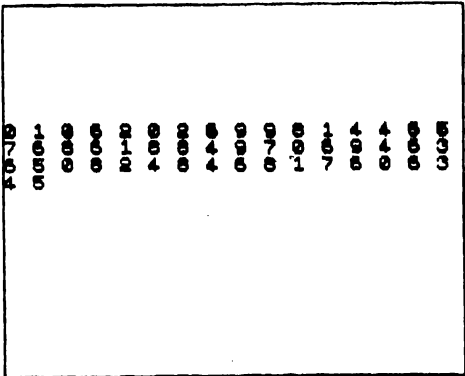


URNA A

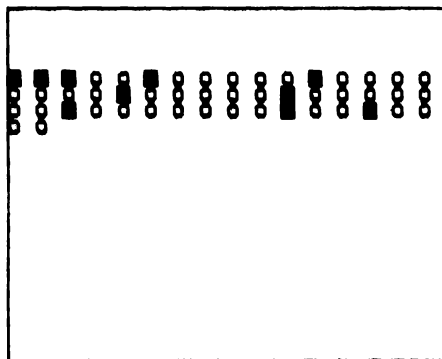


URNA B

- I) con l'elaboratore dobbiamo simulare 50 estrazioni dall'urna B;
- II) Il nostro elaboratore non genera palline bianche o nere, può invece generare numeri. Facciamogli allora costruire un'ipotetica una A (INT(RND*10)).
- III) Esso può simulare estrazioni dall'urna B, se poniamo queste due condizioni:
 - a) i numeri 0 ed 1 rappresentano palline nere;
 - b) gli altri numeri rappresentano palline bianche.
- IV) Questo è il risultato di 50 estrazioni dall'urna A [i numeri casuali generati con INT (RND*10)].



V) Le precedenti estrazioni dall'urna A simulano queste 50 estrazioni dall'urna B



che rappresentano appunto ciò che possiamo vedere stampato sul video mandando in esecuzione il precedente programma.

Se al precedente programma aggiungiamo la seguente istruzione:

```
3 INPUT "QUANTE ESTRAZIONI? "
```

e portiamo la seguente modifica

```
5 FOR I=1 TO N
```

avremo:

```
1 RANDOMIZE
3 INPUT "QUANTE ESTRAZIONI? "
;N
5 FOR I=1 TO N
10 LET A=INT (RND*10)
20 IF A=0 OR A=1 THEN PRINT "■"
30 IF A<>0 AND A<>1 THEN PRINT
"0";" "
40 NEXT I
```

Così operando abbiamo istruito l'elaboratore a simulare "n" estrazioni dall'urna B.

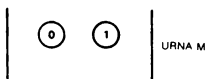
Se istruissimo l'elaboratore a *contare* il numero di volte che ha estratto il numero 0 o il numero 1 (su "n" estrazioni eseguite) potremmo anche istruirlo a calcolare la probabilità teorica, scarto, proprio come abbiamo fatto, con carta e matita, nel capitolo C (ciò sarà mostrato con i programmi dell'ultimo paragrafo).

4.1.6 Lanci di monete e dadi

Lancio di una moneta

Siamo riusciti ad istruire il nostro elaboratore ad eseguire "n" estrazioni da un'urna contenente 8 palline bianche e 2 nere.

Cerchiamo ora di istruire il nostro elaboratore a simulare il lancio di *una moneta*.
Come abbiamo già spiegato, “n” lanci con una moneta possono essere “simulati” da “n” estrazioni dall’urna M.



Dobbiamo allora costruire un programma con il quale l’elaboratore “crea” un’ipotetica urna contenente i numeri 0 ed 1 dalla quale eseguire “n” estrazioni (ogni estrazione simula un lancio).

Dobbiamo inoltre porre queste due condizioni:

- a) se dall’urna M estrae il numero 0 l’elaboratore fa stampare sul video la lettera “T” (nel lancio è apparso TESTA);
- b) se dall’urna M estrae il numero 1 l’elaboratore fa stampare sul video la lettera “C” (nel lancio è apparsa CROCE);

Il programma è il seguente:

```

10 REM LANCI CON UNA MONETA
20 BEEP .1,12
30 INPUT "QUANTI LANCI CON UNA
MONETA? ";N
40 PRINT N;"LANCI CON UNA MONE
TA"
50 PRINT "_____
60 RANDOMIZE
70 FOR I=1 TO N
80 LET A=INT (RND*2)
90 IF A=0 THEN PRINT "T";" "
100 IF A=1 THEN PRINT "C";" "
110 NEXT I

```

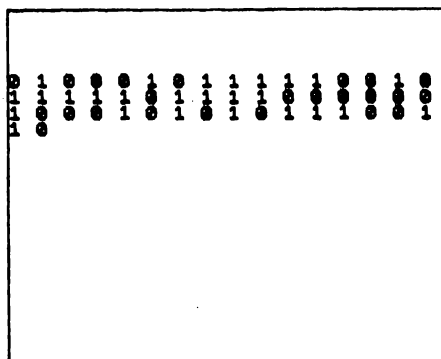
Avendo come punto di riferimento l’ipotetica urna M la linea 30 potrebbe essere così scritta:

```

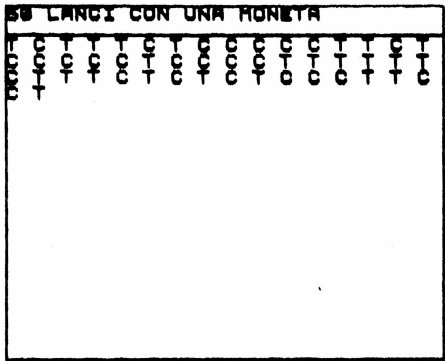
30 INPUT "QUANTE ESTRAZIONI? "
;N

```

Questo è il risultato di 50 estrazioni dall’ipotetica urna M eseguito dall’elaboratore (sono i numeri prodotti dall’istruzione INT(RND*2)):

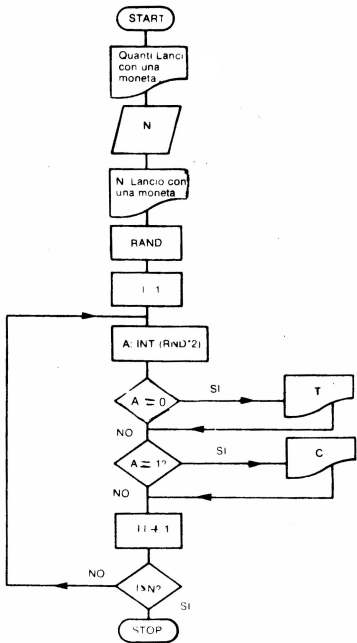


a cui corrisponde questa simulazione di 50 lanci con una moneta (ciò che apparirà sul video):



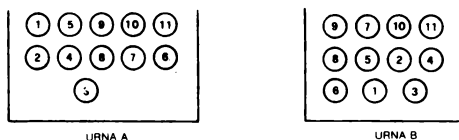
- Analizziamo il programma dal punto di vista del linguaggio informatico:
- a) la variabile I controlla il numero di lanci (o estrazioni dall'urna M);
 - b) la funzione RND genera valori numerici compresi fra 0 e 2 escluso, di cui viene considerata la sola parte intera;
 - c) si abbina il valore numerico 0 al simbolo T (testa) e il valore numerico 1 al simbolo C (croce).

Questo è il corrispondente diagramma di flusso



Lancio di due dadi

Abbiamo già spiegato che il lancio di 2 dadi può essere “simulato” con “n” estrazioni delle seguenti urne numeriche:



Per simulare “n” lanci con due dadi dobbiamo allora scrivere un programma con il quale l’elaboratore viene istruito a ripetere “n” volte queste operazioni:

- “creare” un’ipotetica urna A dalla quale viene eseguita un’extrazione (simulazione lancio di un dado);
- “creare” un’ipotetica urna B dalla quale viene eseguita un’extrazione (simulazione lancio di un dado);
- stampare i numeri estratti sul video e collocarli nelle rispettive urne.

Il programma è il seguente:

```
10 RANDOMIZE
20 FOR I=1 TO 50
30 PRINT INT (RND*6)+1; " "
40 PRINT INT (RND*6)+1,
50 NEXT I
```

Spesso, nel caso del lancio di due dadi, interessa analizzare quali siano gli eventi possibili quando addizioniamo i due numeri che si presentano nei vari lanci.

Per istruire l’elaboratore a eseguire automaticamente anche questa operazione di addizione, il programma precedente diviene:

```
10 REM LANCI CON DUE DADI
15 BEEP .1,12
20 INPUT "QUANTI LANCI CON DUE
DADI?";N
30 PRINT N;"LANCI CON DUE DADI
40 PRINT "
45 RANDOMIZE
50 FOR I=1 TO N
60 LET D=INT (RND*6)+1
70 LET E=INT (RND*6)+1
80 PRINT D;"-";
90 PRINT E;"-->";E+D,
100 NEXT I
```

Mandando in esecuzione questo programma sul video può apparire:

50 LANCI CON DUE DADI															
00	-4	---	>	0											
01	-4	---	>	0											
02	-4	---	>	0											
03	-4	---	>	0											
04	-3	---	>	4											
05	-3	---	>	4											
06	-3	---	>	4											
07	-1	---	>	7											
08	-1	---	>	7											
09	-1	---	>	7											
10	-1	---	>	7											
11	-3	---	>	9											
12	-3	---	>	9											
13	-3	---	>	9											
14	-3	---	>	9											
15	-6	---	>	8											
16	-6	---	>	8											
17	-6	---	>	8											
18	-6	---	>	8											
19	-6	---	>	8											
20	-4	---	>	10											
21	-4	---	>	10											
22	-4	---	>	10											
23	-4	---	>	10											
24	-4	---	>	10											
25	-4	---	>	10											
26	-4	---	>	10											
27	-4	---	>	10											
28	-4	---	>	10											
29	-4	---	>	10											
30	-4	---	>	10											
31	-4	---	>	10											
32	-4	---	>	10											
33	-4	---	>	10											
34	-4	---	>	10											
35	-4	---	>	10											
36	-4	---	>	10											
37	-4	---	>	10											
38	-4	---	>	10											
39	-4	---	>	10											
40	-4	---	>	10											
41	-4	---	>	10											
42	-4	---	>	10											
43	-4	---	>	10											
44	-4	---	>	10											
45	-4	---	>	10											
46	-4	---	>	10											
47	-4	---	>	10											
48	-4	---	>	10											
49	-4	---	>	10											

Vediamo ora di analizzarlo, non più dal punto di vista del modello interpretativo, ma dal punto di vista del linguaggio informatico e delle operazioni che *effettivamente* esegue l'elaboratore:

- alla linea 60 e 70 esistono due istruzioni RND attraverso le quali vengono generati valori numerici casuali compresi fra 1 e 6;
- con l'istruzione 80 diciamo all'elaboratore di stampare sul video il contenuto della memoria di indirizzo D ed accanto ad esso un trattino (—).

Per esempio avremo: 6—

Con l'istruzione 90 diciamo all'elaboratore di stampare sul video il contenuto della memoria di indirizzo E, poiché l'istruzione 80 terminava con un punto e virgola, questo valore numerico verrà stampato accanto al trattino; per cui, ad esempio: 6 — 2

Sempre attraverso l'istruzione 90 vengono stampati: ——— >.

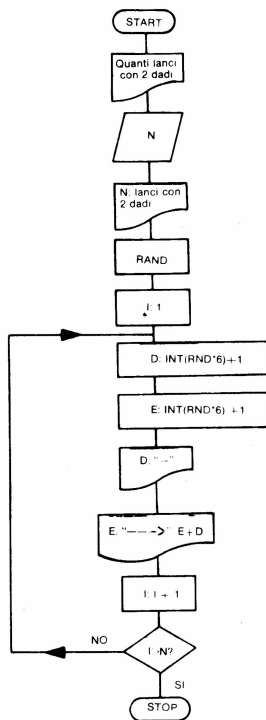
Accanto al simbolo > l'elaboratore stamperà il risultato dell'addizione fra il contenuto numerico della memoria di indirizzo E, e il contenuto della memoria di indirizzo D (nel nostro esempio $6+2=8$).

Sul video apparirà: 6—2 ———> 8

L'istruzione 90 termina con una virgola essa impone di stampare la successiva simulazione del lancio di 2 dadi e il corrispettivo risultato numerico a partire dalla *sedicesima colonna*. Sul video apparirà:

50 LANCI CON DUE DADI															
6	-2	---	>	8											
1	-3	---	>	4											

Questo è il corrispondente diagramma di flusso:



4.1.7 Programmi d'uso

Con questo paragrafo conclusivo ci limitiamo ad illustrare due programmi che possono essere considerati delle simulazioni.

Vanno quindi utilizzati in classe come abbiamo indicato nel primo paragrafo.

Sono programmi abbastanza complessi e quindi non ha senso discuterne con gli alunni. Ma ora essi hanno gli strumenti necessari per comprendere come sia possibile delegare all'elaboratore la soluzione di un certo problema: ad esempio calcolare probabilità, disegnare un istogramma, produrre una tabella, ecc.

Lancio di due monete — tabella — istogramma

```

10 REM LANCI CON DUE MONETE
15 BORDER 7: PAPER 2: INK 7
20 BEEP .1,12
25 PRINT AT 5,0;"VUOI VEDERE L
A SIMULAZIONE DEL LANCIO 2 MONE
TE?"; PRINT AT 9,0;"RISPONDI SI
O NO"
30 INPUT A$: CLS
35 IF A$<>"SI" AND A$<>"NO" TH
EN GO TO 20
  
```

```

28 BEEP .1,12
29 INPUT "QUANTI LANCI CON 2 M
ONETE? ";N
30 PRINT N;" LANCI CON 2 MONET
E": PRINT "

```

```

35 REM VETTORE SIMULAZIONE
37 RANDOMIZE
40 DIM A(4)
50 FOR J=1 TO N
60 LET D=INT (RND*4)+1
65 IF A$="NO" THEN PRINT AT 10
0,"ATTENDI STO LAVORANDO": GO T
O 110
70 IF D=1 THEN PRINT "TT ";
80 IF D=2 THEN PRINT "CT ";
90 IF D=3 THEN PRINT "TC ";
100 IF D=4 THEN PRINT "CC ";
110 LET A(D)=A(D)+1
120 NEXT J
130 PAUSE 200: CLS
160 CLS
170 REM TABELLA
175 PRINT "

```

```

180 PRINT "CASI      FREQ      PER
C      SCARTO","POSSIB  S
PER"
190 PLOT 55,168: DRAW 0,-100
200 PLOT 122,168: DRAW 0,-100
210 PLOT 175,168: DRAW 0,-100
220 PLOT 0,150: DRAW 255,0
225 PLOT 0,128: DRAW 255,0
227 PLOT 0,100: DRAW 255,0
229 PLOT 0,68: DRAW 255,0
230 REM FREQUENZA PROB SPERIM S
CARTO
240 PRINT AT 4,1;"TT":AT 4,10;A
(1);AT 4,18;INT (A(1)/N*100);AT
4,27;INT (A(1)/N*100-25)
250 PRINT AT 7,1;"CC":AT 7,10;A
(4);AT 7,18;INT (A(4)/N*100);AT
7,27;INT (A(4)/N*100-25)
260 PRINT AT 10,1;"TC":AT 10,10
A(2)+A(3);AT 10,18;INT (A(2)+A(
3)/N*100);AT 10,27;INT ((A(2)+A(
3))/N*100-50)
270 PRINT AT 11,1;"CT"
280 PRINT AT 15,10;N;" LANCI"
282 PRINT "

```

```

285 PRINT AT 17,0;"LE PERCENTUA
LI SPERIMENTALI SONO ARROTONDATE
PER DIFETTO"
287 PRINT "

```

```

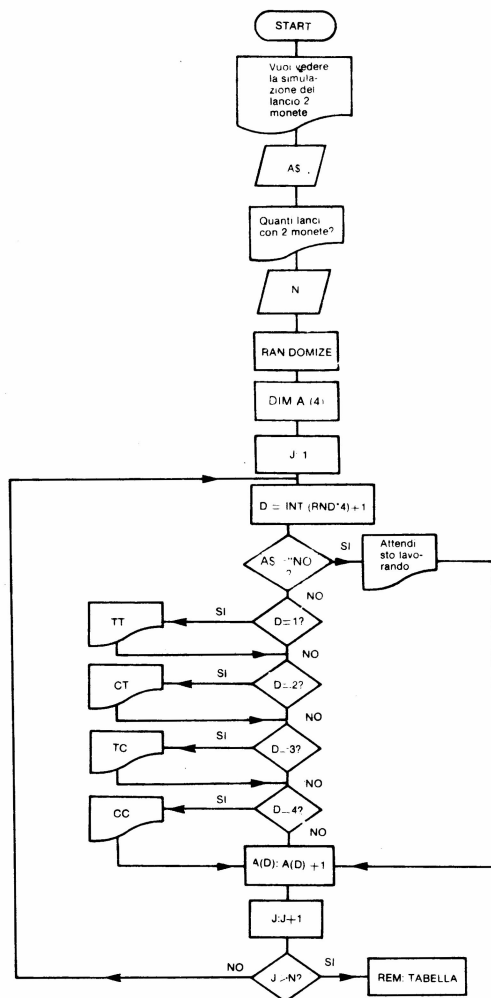
288 FOR I=1 TO 10: BEEP .1,I: N
EXT I
290 FLASH 1: PRINT AT 20,0;"PER
ISTOGRAMMA PREMI UN TASTO";"QUA
LSIASI": FLASH 0
295 IF INKEY$="" THEN GO TO 295
296 CLS
300 REM ISTOGRAMMA
310 PRINT N;" LANCI CON 2 MONET
E"
320 LET X=5
330 FOR J=1 TO 3
340 LET Q=A(J)
345 IF J=2 THEN LET Q=A(2)+A(3)
348 IF J=3 THEN LET Q=A(4)
350 LET S=1
360 IF A(1)/S>18 OR (A(2)+A(3))
/S>18 OR A(4)/S>18 THEN GO TO 600
500 FOR Y=1 TO INT (Q/S)
510 PRINT AT 19-Y,X;"■"
520 NEXT Y
530 LET X=X+5
540 NEXT J
550 PLOT 15,24: DRAW 150,0
560 PRINT AT 20,4;"TT":AT 20,8;
"TC/CT":AT 20,15;"CC"
570 PRINT AT 3,25;"■"=":S
575 FOR I=1 TO 10: BEEP .1,I: N
EXT I
580 REM INCREMENTO VARIABILE S
590 GO TO 1000
600 LET S=S+1: GO TO 360

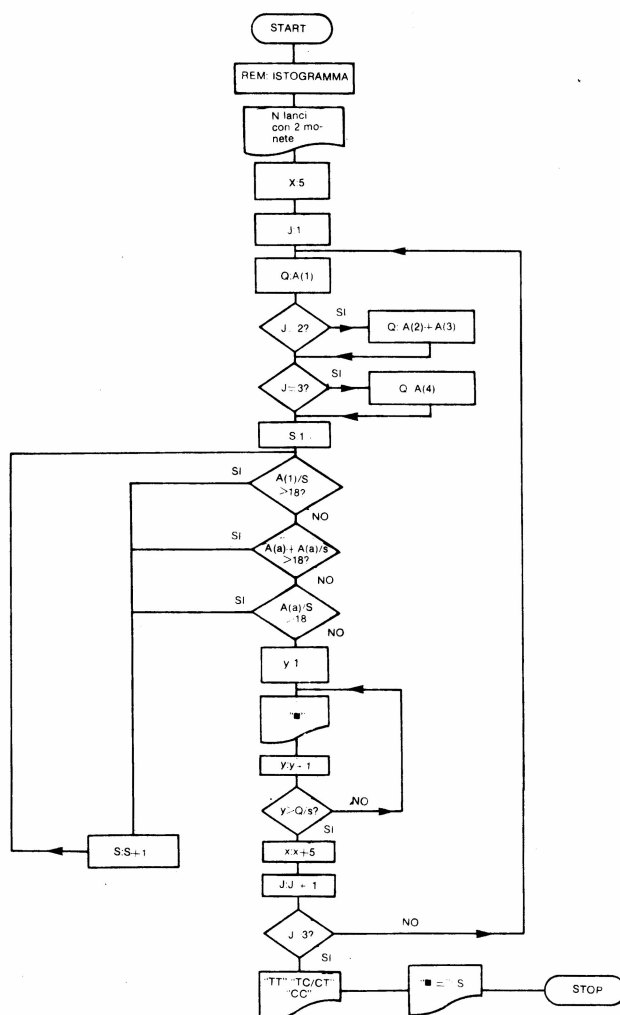
```

Il diagramma di flusso relativo al precedente programma è stato diviso in due parti:

- la prima parte è relativa alle istruzioni comprese fra la 10 e la 170;
- la seconda parte è relativa alle istruzioni comprese fra la 300 e la 600.

La parte del programma compresa fra l'istruzione 175 e l'istruzione 296 è talmente semplice che abbiamo tralasciato di disegnare il relativo diagramma di flusso.





Con questo programma istruiamo l'elaboratore a:

- a) simulare il lancio di due monete;
- b) costruire una tabella ove verranno automaticamente indicati i casi possibili, frequenza, probabilità sperimentale, scarto;
- c) tracciare il relativo istogramma.

Analizziamo ora nel dettaglio le varie parti del programma:

a) Vettore-simulazione

Linea 40: si aprono e si rendono accessibili quattro indirizzi di memoria numerici:



- ciclo FOR/NEXT: si determinano le iterazioni attraverso le quali l'elaboratore esegue la simulazione di N lanci;
- linea 60: ad ogni ciclo FOR/NEXT viene generato un numero casuale compreso tra 1 e 5 escluso; ne viene considerata la sola parte intera;
- linee 70-80-90-100 sul video verrà stampato "TT" se il contenuto della memoria D sarà uno, "CT" se sarà due, "TC" se sarà tre, "CC" se sarà quattro;
- linea 110: con questa istruzione creiamo il *contatore* che ci permetterà di contare il numero di volte che TT, CT, CC, TC ecc. appariranno durante la simulazione di N lanci. Questa istruzione significa: "poni nella memoria di indirizzo A(D) il contenuto della stessa incrementato di un "unità".

Per meglio comprendere cosa avviene nelle memorie dell'elaboratore, immaginiamo che esso abbia simulato 3 lanci, generando attraverso la funzione RND, nei tre cicli iterativi, i numeri 4,1,4.

— Durante il primo ciclo iterativo si avrà questa situazione nelle memorie elaboratore:

J	1
D	4
A(4)	1

e sul video apparirà: CC

— con il secondo ciclo iterativo:

J	2
D	1
A(1)	1

e sul video apparirà: TT

— con il terzo ciclo iterativo:

J	3
D	4
A(4)	2

sul video apparirà: CC

La situazione complessiva delle memorie dopo le 3 iterazioni può essere così schematizzata.

J	1	2	3	4
D	4	1	4	STOP
A(1)	0	1	1	
A(2)	0	0	0	
A(3)	0	0	0	
A(4)	1	1	2	

Abbiamo così costruito un contatore attraverso il quale l'elaboratore può *contare* il numero di volte che TT, TC, CT, CC appaiono durante la simulazione di "N" lanci di due monete, valori numerici che vengono conservati nelle memorie dell'elaboratore.

Tabella

CASI POSSIB	FREQ	PERC SPER	SCARTO
TT	9	18	-7
CC	19	37	12
TC CT	22	44	-6
50 LANCI			
LE PERCENTUALI SPERIMENTALI SONO ARROTONDATE PER DIFETTO			
PER ISTOGRAMMA PREMI UN TASTO QUALSIASI			

Dalla riga 175 alla 229 vi sono le istruzioni con le quali l'elaboratore costruisce l'intelaiatura e l'intestazione della tabella.

Con la linea 240 l'elaboratore viene istruito a stampare:

- 1) nella colonna "casi possibili" TT;
- 2) nella colonna "frequenze" il contenuto della memoria A(1);
- 3) nella colonna "percentuali sperimentali":

frequenza * 100

numero lanci

Al fine di ottenere sempre numeri interi si utilizza la funzione INT (funzione che fornisce la sola parte intera di un numero *arrotondato per difetto*);

- 4) nella colonna "scarti":

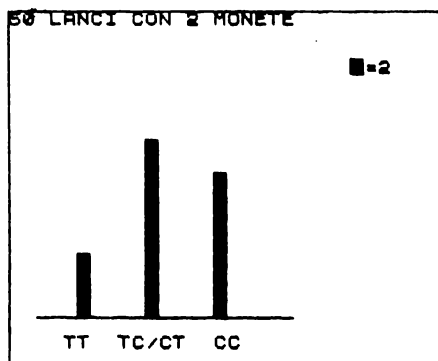
(frequenza x100) — 25,

numero lanci

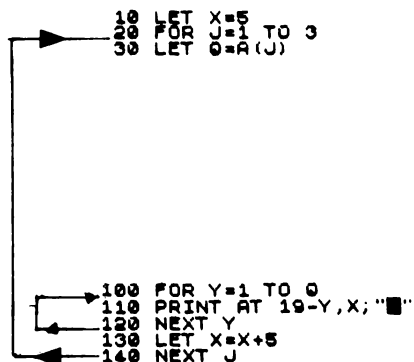
cioè la differenza fra probabilità sperimentale e probabilità teorica (anche qui si utilizza la funzione INT).

La linea 250 è del tutto simile alla precedente; la linea 260 differisce dalle due precedenti perché viene addizionato il contenuto numerico della memoria di indirizzo A(2) con il contenuto numerico della memoria A(3). Si somma così il numero di volte che, durante la simulazione di "N" lanci, è apparso TC e CT.

Istogramma



Consideriamo la doppia iterazione presente in questa ultima parte del nostro programma:



1) con l'iterazione avente "J" come variabile di controllo l'elaboratore leggerà i contenuti numerici delle memorie di indirizzo A(J), valori numerici che verranno posti ad ogni iterazione nella memoria di indirizzo Q. Ad ogni ciclo iterativo il contenuto della memoria di indirizzo X viene incrementato di 5 unità (si posizionano così le 3 colonne istogramma);

2) l'iterazione avente "y" come variabile di controllo viene utilizzata per far disegnare all'elaboratore (nella posizione indicata dalla variabile x) un numero di quadratini corrispondenti al contenuto della memoria di indirizzo Q.

Per meglio comprendere ciò che avviene nelle memorie dell'elaboratore facciamo un esempio pratico: dopo la simulazione di un certo numero di lanci nel vettore di memoria abbiamo i seguenti valori numerici:

A(1)	A(2)	A(3)	A(4)
3	2	2	1

le istruzioni dalla linea 320 sino alla linea 540 determineranno questa situazione negli altri indirizzi di memoria:

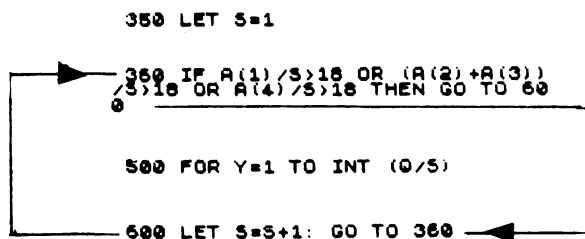
X	5				10					15	20
J	1				2					3	4
Q	3				4				1		STOP
Y	1	2	3	4	1	2	3	4	5	1	2
PRINT "■"	18,5	17,5	16,5	STOP	18,10	17,10	16,10	15,10	STOP	18,15	STOP

Perché alla linea 500 invece di FOR Y=1 TO Q abbiamo scritto FOR Y=1 TO INT (Q/S)?

La pagina video dello SPECTRUM ci permette di costruire istogrammi con colonne formate al massimo da 22 quadratini.

Si tratta allora di far disegnare le colonne istogramma in scala affinché ognuna di esse non sia mai formata da più di 18 quadratini.

Ciò è ottenibile attraverso le seguenti istruzioni:



Per meglio comprendere ciò che avviene nelle memorie dell'elaboratore facciamo un esempio pratico: dopo 200 lanci nel vettore di memoria abbiamo la seguente situazione:

A(1)	A(2)	A(3)	A(4)
50	50	65	35

le precedenti istruzioni determinano questa situazione nell'indirizzo di memoria S date le condizioni poste alla linea 360:

S	1	2	3	4	5	6	7	STOP
A(1)/S	50	25	16,6	12,5	10	8,13	7,1	
A(2)βA(3)/S	115	5,15	38,3	28,75	23	19,1	16,4	
A(4)/S	35	17,5	11,6	8,75	7	5,8	5	

L'elaboratore passerà quindi alla linea 500 solo quando il contenuto numerico della memoria di indirizzo S è 7.

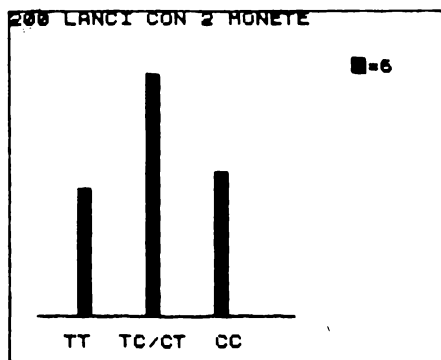
La linea 500, quando S=7, assume allora questo valore:

FOR Y=1 TO INT (50 / 7)

per cui, la colonna istogramma relativa al numero di apparizioni TT durante 200 lanci, sarà formata da 7 quadratini Print.

Riproducendo lo stesso ragionamento quando in J vi sarà il numero 2 e poi il numero 3, capiremo immediatamente che la colonna relativa a "TC/CT" sarà formata da 16 quadratini (115/7), mentre quella relativa a "CC" sarà formata da 5 quadratini Print (35/7).

Sul video apparirà:



LANCIO DI DUE DADI TABELLA - ISTOGRAMMA

```

10 REM LANCI CON 2 DADI
13 BORDER 7: PAPER 2: INK 7
15 BEEP .3,12
20 PRINT AT 5,0;"VUOI VEDERE L
A SIMULAZIONE DEL LANCIO 2 DADI
?" : PRINT AT 9,0;"RISPONDI SI O
NO"
25 INPUT A$: CLS
27 IF A$<>"SI" AND A$<>"NO" TH
EN GO TO 20
28 BEEP .3,12
30 INPUT "QUANTI LANCI CON 2 D
ADI ?":N: IF N=0 THEN GO TO 30
40 REM VETTORE E SIMULAZIONE
45 RANDOMIZE
50 DIM A(11)
60 PRINT N;" LANCI CON 2 DADI"
70 PRINT "

```

```

75 LET U=0
80 FOR J=1 TO N
90 LET D=1+INT (6*AND)+INT (6*
AND)
100 LET A(D)=A(D)+1
105 IF A$="NO" THEN PRINT AT 10
0;"ATTENDI STO LAUORANDO": GO T
O 120
110 PRINT TAB U;D+1;
115 LET U=U+10
116 IF U>30 THEN LET U=0
120 NEXT J
130 PAUSE 200: CLS
140 REM TABELLA
150 PRINT "

```

```

160 PRINT "CASI      FREQ      PER
C SCARTO","POSSIB S
PER"
170 PRINT "

```

```

180 PLOT 60,168: DRAW 0,-120
190 PLOT 122,168: DRAW 0,-120
200 PLOT 175,168: DRAW 0,-120
220 PLOT 0,48: DRAW 253,0
250 REM FREQUENZA PROBABILITA'
SCARTO

```

```

260 FOR D=1 TO 11
270 PRINT AT D+3,0;D+1;AT D+3,1
0;A(D);AT D+3,18;INT (A(D)/N*100
)

```

```

280 LET H=INT (A(D)/N*100)
290 READ Z
300 PRINT AT 3+D,28;H-Z
390 NEXT D
395 DATA 3,6,8,11,14,16,14,11,8
,6,3: RESTORE

```

```

400 PRINT AT 17,0;"LE PERCENTUA
LI SPERIMENTALI SONOARROTONDATE
PER DIFETTO"
410 FOR I=1 TO 10: BEEP .1,I: N
EXT I

```

```

420 FLASH 1: PRINT AT 20,0;"PER
ISTOGRAMMA PREMI UN TASTO","QUA
LUNQUE": FLASH 0

```

```

430 IF INKEY$="" THEN GO TO 430
440 CLS
500 REM ISTOGRAMMA
510 PRINT N;" LANCI"

```

```

520 LET X=3
525 LET S=1
530 FOR J=1 TO 11
540 LET Q=A(J)
550 FOR B=1 TO 11
560 IF A(B)/S>18 THEN GO TO 700
570 NEXT B

```

```

600 FOR Y=1 TO INT (Q/S)
610 PRINT AT 19-Y,X;"■"
615 NEXT Y
620 LET X=X+2
630 NEXT J

```

```

635 PLOT 16,24: DRAW 190,0
640 PRINT AT 19,3;"2";AT 19,5;"
3";AT 19,7;"4";AT 19,9;"5";AT 19
,11;"6";AT 19,13;"7";
650 PRINT AT 19,15;"8";AT 19,17
;"9";AT 19,19;"10";AT 19,21;"11"
;AT 19,23;"12"
660 PRINT AT 3,25;"■"=:"S

```

```

665 FOR I=1 TO 10: BEEP .1,I: N
EXT I

```

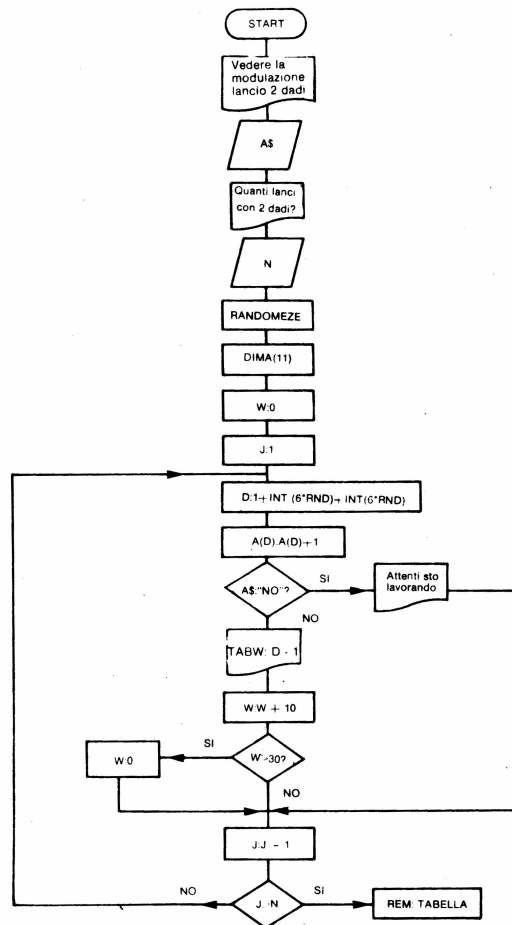
```

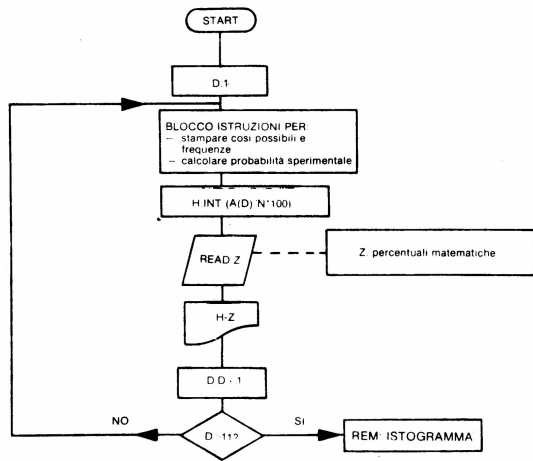
670 GO TO 1000
680 REM INCREMENTO VARIABILE S
700 LET S=S+1: GO TO 550

```

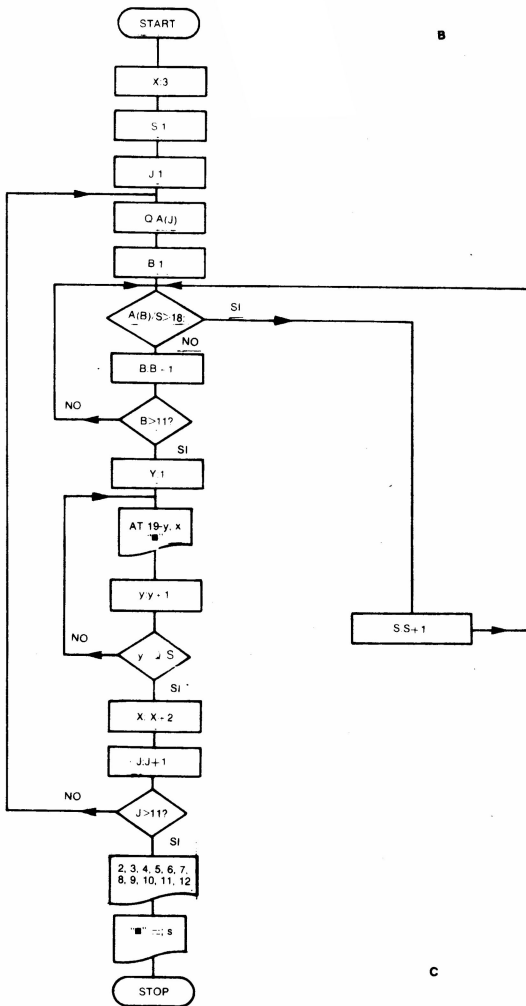
Il diagramma di flusso relativo al precedente programma è stato diviso in 3 parti:

- la prima parte è relativa alle istruzioni comprese fra la linea 10 e la 130 (vettore e simulazione);
- la seconda parte è relativa alle istruzioni comprese fra la linea 140 e la 440 (tabella, frequenze, probabilità, scarto);
- la terza parte è relativa alle istruzioni comprese fra la linea 500 e la 700 (istogramma - incremento variabile).





B



C

Questo programma è molto simile al precedente, ci limiteremo quindi ad analizzarne solo alcuni punti:

- a) linea 50: quando lanciamo 2 dadi abbiamo 11 casi possibili dobbiamo quindi aprire un vettore di memoria di 11 cellette;
- b) linea 300: con questa istruzione, posta all'interno di un ciclo iterativo, facciamo calcolare lo scarto (H-Z). Le varie probabilità teoriche, ad ogni ciclo iterativo, verranno lette alla linea DATA attraverso l'istruzione READ;
- c) linee 550-560-570: attraverso queste istruzioni vengono letti, ad ogni ciclo iterativo, i contenuti numerici del vettore memoria di indirizzo A.

Il contenuto numerico della memoria di indirizzo S verrà incrementata di una unità finché, per tutti i valori numerici contenuti nel vettore di memoria, si avrà:

$$A(B)/S > 18$$

4.2 I quadrilateri

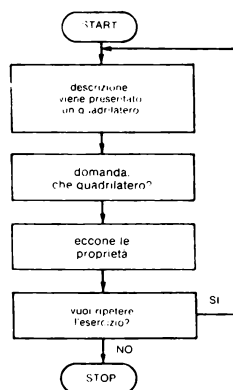
PIERO D'ALFONSO E GIOVANNI MARTORANA

L'unità che qui si presenta è un piccolo frammento di un testo di matematica per la scuola media.

Viene proposta come dimostrazione di quella forma di istruzione assistita da computer che comunemente viene chiamata "libro elettronico".

Esso rientra nel capitolo che riguarda l'insieme dei quadrilateri.

La strategia di questo frammento si compone di un ciclo descrizione-domande-descrizione ripetibile più volte secondo il diagramma:



nel quale l'esecuzione dell'esercizio intermedio (che quadrilatero è?) è un filtro posto tra la presentazione del quadrilatero e la descrizione delle sue proprietà.

Si vuole che l'alunno abbia sott'occhio una sola figura, si impegni a definire il nome esatto prima che l'elaboratore vi associ le rispettive proprietà. Con questa unità non si

verifica se la risposta esatta rappresenti la comprensione *definitiva* della relazione tra forma della figura e sua appartenenza al sottoinsieme dei quadrilateri che le è proprio. Vi è solo l'intenzione di concordare l'associazione tra forma della figura, dati che la connotano e nome del sottoinsieme.

Un'altra unità si dovrà associare a questa per farsi carico delle verifiche, formative e sommative.

La I^a parte è descritta in modo analitico perché può essere utilizzata (e dovrebbe esserlo) in modo vantaggioso per costruire delle singole U.D.S..

L'insegnante che utilizzasse questo programma è invitato a costruire personalmente, e a far costruire dagli alunni, singoli quadrilateri attraverso programmi in BASIC, prima con valori fissi e poi con valori RND.

Si tratta di poche righe di facile scrittura ma di notevole significato per la comprensione dell'uso degli assi cartesiani e delle regole geometriche.

4.2.1 I^a Parte. Descrizione

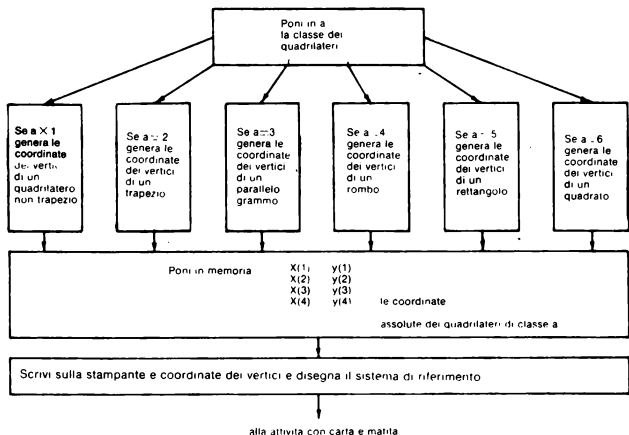
La descrizione/presentazione del quadrilatero si compone a sua volta di tre parti in sequenza:

ELABORATORE	ALUNNO	ELABORATORE
generazione delle coordinate e loro presentazioni	attività con carta e matita	visualizzazione del quadrilatero sul video

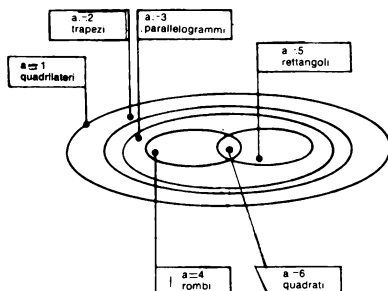
Analizziamoli distintamente.

Generazione di coordinate

Per generare le coordinate di un quadrilatero ci si è appoggiati allo schema che segue:



Le classi dei quadrilateri sono ovviamente determinate col criterio insiemistico che considera l'insieme dei trapezi incluso in quello dei quadrilateri, l'insieme dei parallelogrammi incluso in quello dei trapezi, i due insiemi dei rombi e dei rettangoli intersecati tra di loro e inclusi in quello dei parallelogrammi, con l'insieme dei quadrati intersezione tra rombi e rettangoli.



Non sono stati presi in considerazione come classe i deltoidi che risultano pertanto inclusi nella fascia esterna dei quadrilateri.

Per distinguere questa fascia dalle altre in modo più connotato si è ritenuto opportuno definirla, ed è assolutamente corretto, come la classe dei "quadrilateri non trapezi".

La distinzione tra le varie classi così delimitate è fornita al computer attraverso l'indirizzo di memoria a.

Il programma principale

Il computer deve generare dunque in modo casuale un numero compreso tra 1 e 6 e inserirlo nell'indirizzo di memoria a, secondo l'istruzione alla linea 1020

```
1020>LET a=INT (RND*6) +1
```

Successivamente dovrà stabilire un collegamento separato con i sottoprogrammi che generano le coordinate dei vertici delle varie classi dei quadrilateri:

```
1023>GO SUB 1000+100*a: REM
```

Da questa istruzione appare evidente un vincolo.

Il programma che genera le coordinate dei vertici di un quadrilatero non trapezio dovrà incominciare alla linea 1100 e finire entro la linea 1199 per non interferire con quello dei trapezi.

Il programma che genera le coordinate dei vertici di un trapezio dovrà incominciare alla linea 1200 e finire entro la linea 1299.

E così via fino al programma del quadro che dovrà incominciare alla linea 1600 e finire entro la linea 1699.

Le linee da 1000 a 1099 sono riservate dunque per operazioni di tipo generale, cioè valide per tutti i sottoprogrammi; oppure per smistare ai vari sottoprogrammi. È questo il motivo per cui di qui in poi queste linee verranno definite del *programma principale*. Eccone la lista associata alla distinzione delle varie parti:

```

1000>REM PROGRAMMA PRINCIPALE
1001 DIM x(4): DIM y(4): DIM a$(
4,1): DIM a(4): DIM c(4)
1002 RESTORE
1003 FOR i=1 TO 4: READ a$(i): N
EXT i: REM legge nomi vertici
1004 CLS: PRINT AT 3,0;"NOTA BE
NE. IN QUESTO PROGRAMMA": PRINT
: PRINT "I NUMERI CHE DEFINISCON
O LE": PRINT: PRINT "MISURE DEG
LI ANGOLI SONO": PRINT: PRINT "
RIFERITI AL GRADO."
1005 PRINT: PRINT "I NUMERI CHE
DEFINISCONO LE": PRINT: PRINT
"COORDINATE SONO RIFERITI A": P
RINT: PRINT "PIXEL CIOE' AI PUN
TI DEL VIDEO."
1006 INPUT "Stai incominciando u
na nuova unita' oppure vuoi rich
iamare i da-ti (dati=d / altri=
enti ENTER)";ss: IF ss="d" THEN
GO SUB 4000
1007 IF ss="d" THEN GO TO 1027
1008 REM GENERA LE FIGURE
1009 LET a=INT (RND*6)+1
1010 REM ALLE VARIE SUBROUTINES
1011 GO SUB 1000+100*a: REM cors
truzione random figure
1012 FOR i=1 TO 4: IF x(i)>=251
OR y(i)>=171 THEN GO TO 1023: RE
M evita uscita da video
1013 NEXT i
1014 GO SUB 9700: REM coordinate
e sist. di riferimento
1015 GO SUB 1700: REM opzioni

1016 IF z$="s" OR k$="s" THEN GO
SUB 2070
1017 GO SUB 9800
1018 IF z$="n" AND k$="n" THEN G
O SUB 2000
1019 IF z$="n" AND k$="n" THEN G
O SUB 1800
1020 IF z$="n" AND k$="n" THEN G
O SUB 9800: REM

1021 GO SUB 8000: REM

1022 REM PROCEDURA DI ITERAZIONE
DEL PROGRAMMA
1023 CLS: PRINT AT 10,0;"Vuoi r
iprovare? (si=s / no=n)"
1024 IF INKEY$("<") THEN GO TO 10
45
1025 IF INKEY$("<")="s" AND INKEY$("<")
="n" THEN GO TO 1047
1026 IF INKEY$=" " THEN GO TO 104
8
1027 IF INKEY$("<")="n" THEN GO TO 1
090
1028 CLS: INPUT "Innanzitutto,
se hai il microdrive, estrai il
cartridge. Stacca l'alimentatore
prima di spegne-re il video e
la stampante":x$
1029 PRINT AT 10,11;"F I N E"
1030 GO TO 1080
1031 GO TO 1004
1032 STOP

```

Organizzazione sul video degli elementi da visualizzare

Il video è stato diviso in tre aree (o finestre) in modo da evitare interferenze o sovrapposizioni tra disegni, dati e spiegazioni o domande.

FINESTRA DEI DATI
(15 righe di 8 caratteri)

FINESTRA DEI DISEGNI

FINESTRA DELLE SPIEGAZIONI O DELLE ISTRUZIONI
O DELLE DOMANDE (7 righe di 32 caratteri)

Alla finestra dei dati si accede con le istruzioni PRINT. Essa ha disposizione 15 righe di 8 caratteri.

Alla finestra delle spiegazioni e delle domande si accede con le istruzioni INPUT ed essa ha a disposizione 7 righe di 32 caratteri.

Alla finestra dei disegni si accede con le istruzioni di PLOT, DRAW e PRINT AT (per lettere e numeri).

Questa finestra ha a disposizione un'area di 192x120 pixel per il disegno delle varie figure, a partire da $x=64$ fino a $x=255$ per le ascisse e da $y=56$ fino a $y=175$ per le ordinate. Di questa area una striscia a sinistra e in basso è dedicata alla rappresentazione delle coordinate cartesiane e delle relative misure.

Questa striscia occupa uno spessore di 16 pixel in verticale e di 14 pixel in orizzontale di modo che l'area effettivamente occupabile dai disegni dei quadrilateri parte da $x=80$ per le ascisse e da $y=70$ per le ordinate. Questo implica che le coordinate dei vertici delle figure abbiano misure relative a questo sistema di riferimento e non a quello dello SPECTRUM.

Si dà, su carta millimetrata, la finestra dei disegni con l'avvertimento che, per consentire una ricostruzione maggiormente semplificata, i valori a disposizione della finestra ($255-80=175$ per le x e $175-56=119$ per le y) sono stati divisi per due e arrotondati al valore intero più basso.

Con ciò la scala graduata delle ascisse va da 0 a 87 (parte intera di $175/2$) e quella delle ordinate da 0 a 59 (parte intera di $119/2$).

Questa spiegazione era necessaria per consentire la comprensione dei limiti imposti dal video nella generazione delle coordinate dei vari quadrilateri.

Dimensionamento delle memorie

Ad ogni vertice da generare saranno dunque associati cinque valori ognuno dei quali dovrà essere conservato in un indirizzo proprio per l'intera unità didattica. Chiamato (i) un vertice qualsiasi i cinque indirizzi sono:

$x(i)$ ascissa assoluta cioè riferita alle coordinate dello Spectrum;

$y(i)$ ordinata assoluta cioè riferita alle coordinate dello Spectrum;

$a\$ (i)$ lettera dell'alfabeto associata al singolo vertice;

$a(i)$ ascissa relativa cioè riferita alle coordinate della finestra dei disegni;

$c(i)$ ordinata relativa cioè riferita alla coordinata della finestra dei disegni.

Entro i limiti della finestra ed entro i vincoli imposti dalle condizioni di appartenenza alla classe dei quadrilateri attribuita da a , verranno generate con valori casuali le coordinate assolute $x(i)$ e $y(i)$. Le coordinate $a(i)$ e $c(i)$ dipenderanno da quelle a meno del valore fisso imposto dalla posizione della finestra divisa per due.

In $a\$ (i)$ verrà posta una sola lettera (il nome del vertice) valida per l'intera unità didattica con la funzione di istruire in modo permanente la corrispondenza tra indice e vertice:

1	→	M
2	→	N
3	→	P
4	→	Q

e di permettere la lettura delle coordinate di un vertice scritto in lettera:

M	→	x(1)
		y(1)
N	→	x(2)
		y(2)
P	→	x(3)
		y(3)
Q	→	x(4)
		y(4)

Il fatto che queste lettere sono identiche per qualunque quadrilatero suggerisce di porle in una linea di DATA.

```
9900>DATA "M","N","P","Q"
```

La linea di dimensionamento prevede in conclusione quattro indici (uno per ogni vertice) per tipo di memoria come risulta alla linea 1003:

```
1003>DIM x(4): DIM y(4): DIM a$(  
4,1): DIM c(4): DIM c(4)
```

dove ovviamente il valore 1 in DIM a\$(4,1) indica che la singola stringa di a\$ sarà composta di un solo carattere in tutti e quattro i casi.

L'istruzione che regola la dipendenza di a(i) da x(i) e di c(i) da y(i) è alla linea 9701:

```
9701>FOR i=1 TO 4: LET a(i)=INT  
((x(i)-80)/2): LET c(i)=INT ((y(  
i)-70)/2): NEXT i
```

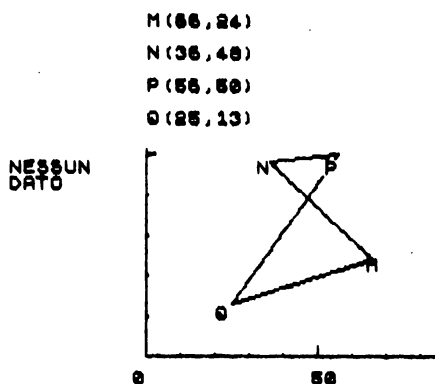
Le istruzioni che richiamano inizialmente e permanentemente le lettere corrispondenti ai vertici sono alle linee 1005 e 1006.

```
1005>RESTORE  
1006 FOR i=1 TO 4: READ a$(i): N  
EXT i: REM legge nome vertici
```

Quadrilatero non trapezio

Le coordinate generate casualmente con la funzione RND per creare un quadrilatero non trapezio hanno un margine di variabilità per x da 91 a 251 e per y da 85 a 171 (in valori assoluti).

Non vi sono altri vincoli neppure sull'ordine di generazione dei vertici. Ciò rende possibile la costruzione di quadrilateri con lati intersecati tra di loro come si vede nel disegno che segue.



Si è ritenuto opportuno conservare questa possibilità per mettere a disposizione degli insegnanti e degli allievi delle occasioni di ragionamento sul rispetto della successione dei vertici nel tracciamento delle figure.

Infatti per le classi di quadrilateri incluse nei trapezi il rispetto dell'ordine dei vertici nella costruzione dei lati è tassativo. Basti riflettere alle condizioni che permettono di conservare il parallelismo dei lati opposti per rendersene conto.

```
1100>REM QUADRILATERO NON TRAPEZ
110
1110 FOR i=1 TO 4
1120 LET X(i)=INT (RND*80)*2+90:
LET Y(i)=INT (RND*43)*2+86
1130 NEXT i
1190 RETURN
```

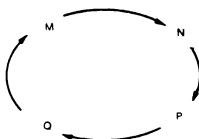
Trapezi

La cattiva risoluzione determinata anche dalla riduzione dello spazio per il disegno rispetto all'intero video impedisce di generare figure ruotate rispetto all'orizzontale o verticale.

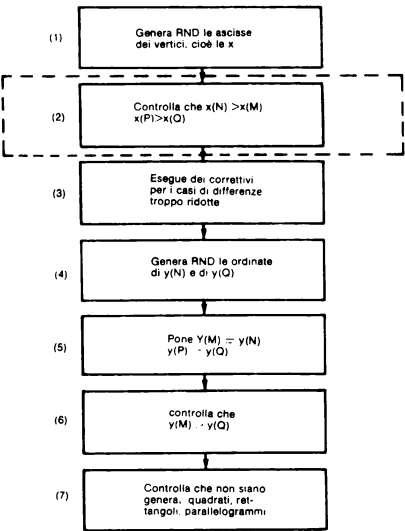
Infatti così facendo è troppo elevato il numero dei casi in cui (data la generazione RND delle coordinate dei vertici) le basi dei trapezi diventano irriconoscibili a causa della discontinuità del tratto. Pertanto i vertici dei trapezi verranno generati in modo che i due lati paralleli siano sempre orizzontali.

È questo un limite oggettivo dal punto di vista didattico perché induce (o comunque non corregge) il possibile convincimento che una condizione dell'appartenenza alla classe dei trapezi sia l'orizzontalità.

La successione dei vertici è oraria a partire da quello più in alto a sinistra:



Il programma che ne deriva ha il seguente schema:



L'operazione (1) ha il compito di generare le ascisse di M, N, P, Q. I numeri RND che verranno generati avranno un valore minimo di 100 e massimo di 250. E cioè:

$$100 \leq x(M) \text{ or } x(N) \text{ or } x(P) \text{ or } x(Q) \leq 250$$

La generazione random delle x è spezzata in due parti: (linea 1210) per x(M) e x(N) e linea 1216 per x(P) e x(Q) per agevolare le operazioni di controllo.

L'operazione (2) è opzionale. Serve ad evitare la generazione di trapezi intrecciati. Essa è eseguita considerando le corrispondenze tra gli indirizzi di memoria x(i), y(i) e le coordinate dei vertici M, N, P, Q.

Dunque

$$\begin{array}{l} x(N) \longrightarrow x(2) > x(1) \longrightarrow x(M) \\ x(P) \longrightarrow x(3) > x(4) \longrightarrow x(Q) \end{array}$$

I correttivi di cui all'operazione (3) sono di due tipi: l'uno tendente ad evitare che venga generata una figura con una base di valore 0 ($M \equiv N$ o $P \equiv Q$), l'altro tendente ad evitare che venga generata una figura con altezza pari a 0 (MN giacente sulla stessa retta di PQ).

L'operazione (4) ha il compito di generare le ordinate di N e Q. I numeri RND che verranno generati avranno un valore minimo di 86 e massimo di 170. Dunque:

$$86 < y(N) \text{ or } y(Q) < 170$$

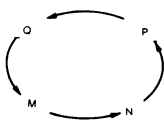
L'operazione (5) completa la tabella:

x(M)	x(1) =	generata casualmente
x(N)	x(2) =	" "
y(N)	y(2) =	" "
x(P)	x(3) =	" "
x(Q)	x(4) =	" "
y(Q)	x(4) =	" "
y(M)	y(1) =	y(2)
y(P)	y(3) =	y(4)

L'operazione (6), non essenziale, serve a impedire che il verso di successione dei vertici sia antiorario. Per far ciò basta garantirsi che sia sempre:

$$y(M) \quad y(1) > y(4) \quad y(Q)$$

Se così non fosse infatti si verificherebbe la situazione:



L'operazione (7) esclude che vengano generati trapezi particolari quali sono il parallelogramma, il rettangolo, il quadrato; ciò comporta verificare la condizione:

$$x(M) - x(Q) \quad x(1) - x(4) \neq x(2) - x(3) \quad x(N) - x(P)$$

Il sottoprogramma completo per generare i vertici di un trapezio è:

```

1200>REM TRAPEZIO
1210 FOR i=1 TO 2: LET x(i)=INT
(RND*75)*2+100: NEXT i
1214 IF x(2)-x(1)<=32 THEN LET x
(2)=x(2)+20
1216 FOR i=3 TO 4: LET x(i)=INT
(RND*75)*2+100: NEXT i
1219 IF x(3)-x(4)<=32 THEN LET x
(3)=x(3)+20
1220 FOR i=2 TO 4 STEP 2: LET y(
i)=INT (RND*43)*2+84: LET y(i-1)
=y(i): NEXT i
1222 IF y(4)>y(1) THEN GO TO 122
0
1224 IF y(1)-y(4)<=32 THEN LET y
(1)=y(1)+20: LET y(2)=y(1)
1226 IF x(1)=x(4) AND x(2)=x(3)
THEN GO TO 1200
1228 IF x(1)-x(4)=x(2)-x(3) THEN
GO TO 1200
1290 RETURN
  
```

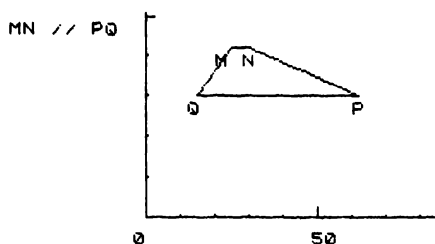
dove con la linea 1290 si ritorna al programma principale.

Da questo si ottengono, per esempio, valori di M, N, P, Q come i seguenti:

```

M(25,42)
N(30,42)
P(62,30)
Q(15,30)
  
```

a cui corrisponde la figura:

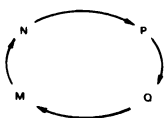


Parallelogrammi

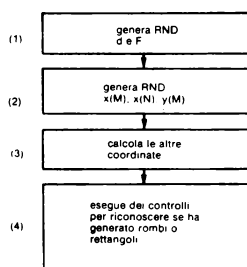
Anche per i parallelogrammi (come per i sottoinsiemi dei rombi, dei rettangoli e dei quadrati) vale il ragionamento fatto sui vincoli imposti dalla cattiva risoluzione del video e più in particolare della finestra dei disegni.

Per questo motivo si resta vincolati alla generazione di parallelogrammi con due lati opposti orizzontali.

La successione di vertici è oraria a partire dal primo in basso a sinistra.



Delle otto coordinate da definire ne vengono generate 3 con valori casuali e ad esse sono associati due parametri (d e f), sempre casuali secondo lo schema.



Operazione (1). Il parametro d è un numero casuale che ha un valore minimo di 60 e un massimo di 90.

Esso definisce il valore del lato di base del parallelogramma.

Il parametro f è anch'esso un numero casuale con un minimo di 16 e massimo di 56. Esso definisce il valore dell'altezza del parallelogramma.

Operazione (2). La $x(1)$, ascissa di M, ha un valore minimo di 100 e massimo di 120. La $x(2)$, ascissa di N, ha lo stesso intervallo ed è generata separatamente dalla $x(1)$.

La $y(1)$, ordinata di M, ha un valore minimo di 82 e massimo di 106.

Operazione (3). È quella che permette di completare il quadro:

$x(M) \longrightarrow$	$x(1) =$	generata casualmente
$y(M) \longrightarrow$	$y(1) =$	" "
$x(N) \longrightarrow$	$x(2) =$	" "
$y(N) \longrightarrow$	$y(2) =$	$y(1) + f$
$x(P) \longrightarrow$	$x(3) =$	$x(2) + d$
$y(P) \longrightarrow$	$y(3) =$	$y(2)$
$x(R) \longrightarrow$	$x(4) =$	$x(1) + d$
$y(Q) \longrightarrow$	$y(4) =$	$y(1)$

I controlli di cui all'operazione (4) avvengono in questo modo.

Per riconoscere che si è generato un rettangolo basta inserire un parametro q che da 0 passi a 1 se risultano uguali i valori di N e assegnare in quel caso il valore 5 ad a .

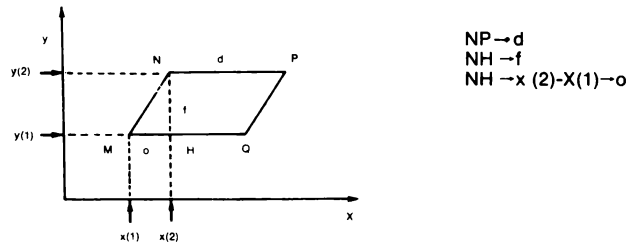
```

1305>LET q=0
1355>IF x(1)=x(2) THEN LET q=1
1365>IF q=1 AND a<>4 THEN LET a=
5

```

Per riconoscere che si è generato un rombo è sufficiente verificare se il quadrato della differenza delle ascisse di N e M, sommato al quadrato del parametro f è uguale al quadrato del parametro d . In tal caso si porrà $a = 4$.

Infatti osservando la figura:



perché MNPQ sia un rombo occorre che:

$MN = NP$

Ma $NP=d$ e MN è l'ipotenusa del triangolo rettangolo MNH. Posti allora:

$o=x(2)-x(1)$

$f=y(2)-y(1)$

valgono le condizioni della linea:

```

1357>LET o=x(2)-x(1): IF o*o+f*f
=d*d THEN LET a=4

```

Per riconoscere se si è generato un quadrato basta verificare se $q=1$ e se $a=4$. In quel caso essendo la figura sia rombo che rettangolo, si tratterà appunto di un quadrato e sarà $a=6$:

```
1360>IF q=1 AND a=4 THEN LET a=6
```

Il sottoprogramma completo per generare i vertici di un parallelogramma è dunque:

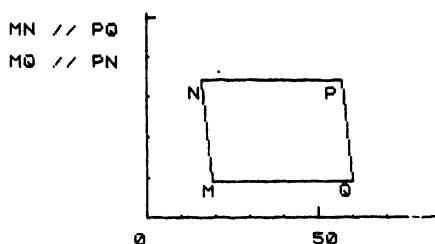
```
1300>REM PARALLELOGRAMMA
1305 LET q=0
1310 LET d=INT (RND*15)*2+60: LE
T X(1)=INT (RND*10)*2+100: LET X
(4)=X(1)+d
1320 LET X(2)=INT (RND*12)*2+100
: LET X(3)=X(2)+d
1330 LET f=INT (RND*20)*2+16: LE
T Y(1)=INT (RND*12)*2+82
1335 LET Y(4)=Y(1)
1340 LET Y(2)=Y(1)+f: LET Y(3)=Y
(2)
1355 IF X(1)=X(2) THEN LET q=1
1357 LET o=X(2)-X(1): IF o*o+f*f
=d*d THEN LET a=4
1360 IF q=1 AND a=4 THEN LET a=6
1365 IF q=1 AND a<>4 THEN LET a=
5
1390 RETURN
```

dove con la linea 1390 si ritorna al programma principale.

Da questo sottoprogramma sono ottenibili valori come i seguenti:

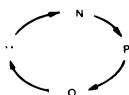
```
M(19,9)
N(16,34)
P(57,34)
Q(60,9)
```

a cui corrisponde la figura:

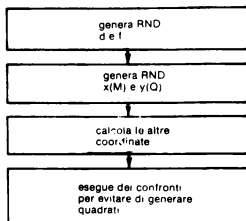


Rombi

I vincoli imposti dalla cattiva risoluzione del video suggeriscono di generare figure di rombi con le diagonali l'una verticale e l'altra orizzontale. La successione dei vertici è oraria a partire dal vertice più a sinistra:



Delle otto coordinate da definire ne vengono generate 2 con valori casuali e ad esse sono associati due parametri (d e f), sempre RND, secondo lo schema:



Operazione (1). Il parametro f è anch'esso un numero RND con minimo di 8 e massimo di 60. Esso definisce il valore della semidiagonale orizzontale. Il parametro f è anch'esso un numero RND con minimo di 8 e massimo di 40. definisce il valore della semidiagonale verticale.

Operazione (2). La x(1), ascissa di M, ha un valore minimo di 80 e massimo di 120. La y(4), ordinata di Q, ha un valore minimo di 80 e massimo di 100.

Operazione (3) Permette di completare il quadro:

x(M)	→	x(1)	=	generata casualmente
y(Q)	→	y(4)	=	" "
y(M)	→	y(1)	=	y(4)+f
x(Q)	→	x(4)	=	x(1)+d
x(N)	→	x(2)	=	x(4)
y(N)	→	y(2)	=	y(1)+f
x(P)	→	x(3)	=	x(2)+d
y(P)	→	y(3)	=	y(1)

Operazione (4). Per riconoscere se si è generato un quadrato basta verificare se $d=f$ allora si porrà $a=6$:

```
1465>IF d=f THEN LET a=6
```

Il sottoprogramma completo per generare i vertici di un rombo è dunque:

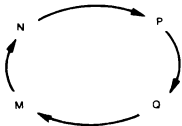
```

1400>REM ROMBO
1410 LET d=INT (RND*26)*2+8: LET
x(1)=INT (RND*20)*2+80: LET x(4
)=x(1)+d
1420 LET x(2)=x(4): LET x(3)=x(2
)+d
1440 LET f=INT (RND*16)*2+8: LET
y(4)=INT (RND*10)*2+80: LET y(1
)=y(4)+f
1450 LET y(3)=y(1): LET y(2)=y(1
)+f
1465 IF d=f THEN LET a=6
1490 RETURN
  
```

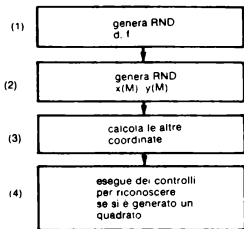
Vengono così generate coordinate e relativa Figura, analoga a quelle precedenti.

Rettangoli

Per gli stessi motivi espressi per i precedenti quadrilateri i vertici dei rettangoli sono generati in modo da formare lati orizzontali o verticali. La loro successione è oraria a partire dal primo in basso a sinistra:



Delle otto coordinate da definire ne vengono generate 2 con valori casuali e ad esse sono associati due parametri (d e f), sempre RND, secondo lo schema:



Operazione (1). Il parametro d è un numero con un minimo di 60 e un massimo di 100 e rappresenta il valore del lato di base.

Il parametro f ha un minimo di 40 e un massimo di 60 e rappresenta il valore di un lato perpendicolare alla base.

Operazione (2). La x(1), ascissa di M, ha un minimo di 111 e un massimo di 141. La y(1), ordinata di M, ha un minimo di 81 e un massimo di 96.

Operazione (3) Permette di completare il quadro:

x(M)	→	x(1)	=	generata casualmente
y(M)	→	y(1)	=	" "
x(N)	→	x(2)	=	x(1)
y(N)	→	y(2)	=	y(1)+f
x(P)	→	x(3)	=	x(4)
y(P)	→	y(3)	=	y(2)
x(Q)	→	x(4)	=	x(1)+d
y(Q)	→	y(4)	=	y(1)

Operazione (4) Si limita ad imporre che a=6 se risulta f=d come per il rombo. È evidente che la differenza tra le due condizioni è rappresentata dagli elementi a cui i parametri si applicano. In questo caso sono i lati del rettangolo, nel caso del rombo sono le semidiagonali.

Il sottoprogramma completo per generare i vertici di un rettangolo è dunque:

```

1500>REM RETTANGOLO
1510 LET d=INT (RND*20)*2+60: LE
T x(1)=INT (RND*15)*2+112: LET x
(4)=x(1)+d
1520 LET x(2)=x(1): LET x(3)=x(4
)
1530 LET f=INT (RND*10)*2+40: LE
T y(1)=INT (RND*8)*2+80
1540 LET y(2)=y(1)+f: LET y(4)=y
(1): LET y(3)=y(2)
1545 IF d=f THEN LET a=6
1550 RETURN

```

Vengono così generate coordinate e relativa figura.

Quadrati

La generazione delle coordinate dei vertici del quadrato non si discosta da quella del rettangolo se non perché f e d sono uguali e quindi viene generato solo d. Il quadro completo è:

x(M)	→	x(1)	=	generata casualmente
y(M)	→	y(1)	=	" "
x(N)	→	x(2)	=	x(1)
y(N)	→	y(2)	=	y(1)+d
x(P)	→	x(3)	=	x(4)
y(P)	→	y(3)	=	y(2)
x(Q)	→	x(4)	=	x(1)+d
y(Q)	→	y(4)	=	y(1)

Il sottoprogramma completo per generare i vertici di un quadrato è dunque:

```

1600>REM QUADRATO
1610 LET d=INT (RND*20)*2+60: LE
T x(1)=INT (RND*15)*2+112: LET x
(4)=x(1)+d
1620 LET x(2)=x(1): LET x(3)=x(4
)
1630 LET y(1)=INT (RND*12)*2+82
1640 LET y(2)=y(1)+d: LET y(4)=y
(1): LET y(3)=y(2)
1670 RETURN

```

Vengono così generate coordinate e relativa figura.

Scrittura sul video delle coordinate

Una volta generate le coordinate e conservate nei rispettivi indirizzi di memoria il computer riceve l'istruzione di tornare al programma principale (RETURN). Di lì, dopo un ultimo controllo che impedisce per l'ascissa valori superiori, a 250 e per l'ordinata a 170, viene inviato alla subroutine di visualizzazione delle coordinate e rappresentazione del sistema di riferimento.

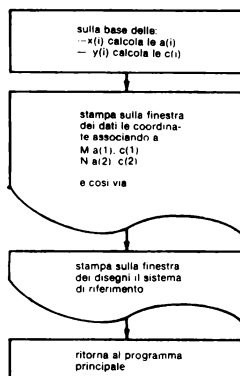
Le linee del programma principale sono:

```

1024>FOR i=1 TO 4: IF x(i)>251
OR y(i)>171 THEN GO TO 1023: RE
M evita uscita da video
1025 NEXT i
1027 GO SUB 9700: REM coordinate
e sist. di riferimento

```

dove la linea 1023 a cui rimanda l'iterazione delle linee 1024 — 1025 è quel GOSUB 1000+100*a che inviava, a seconda del valore contenuto in a, alle subroutine delle singole classi di quadrilateri per la generazione delle coordinate dei vertici. È questo l'ultimo controllo prima di procedere col programma alla linea 1027 che invia alla subroutine che fornisce le coordinate e il sistema di riferimento. In questa subroutine viene realizzato il seguente lavoro:



Il programma completo è il seguente:

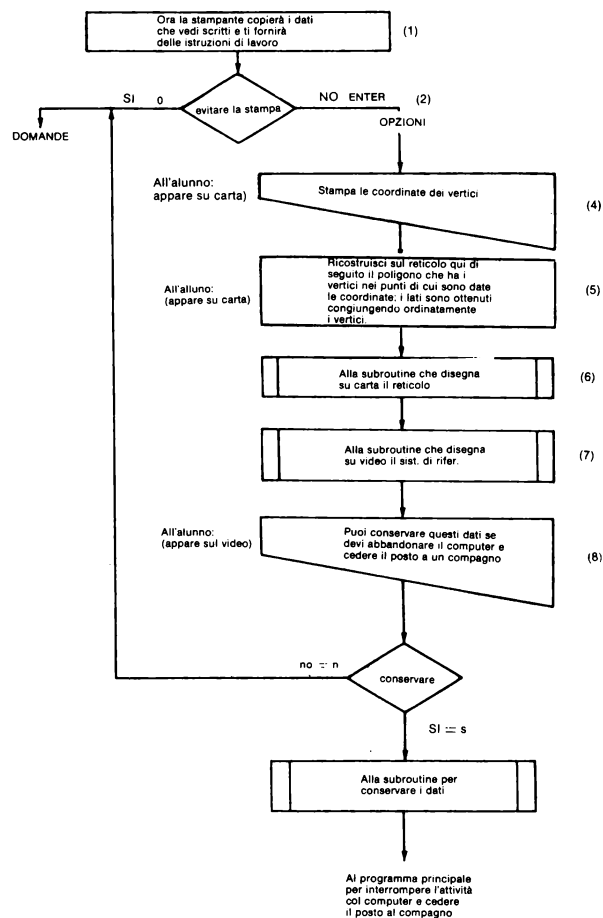
```

9700>REM FORNISCE LE COORDINATE
E IL SISTEMA DI RIFERIMENTO
9701 FOR i=1 TO 4: LET a(i)=INT
((X(i)-80)/2): LET c(i)=INT ((y(
i)-70)/2): NEXT i
9702 CLS: FOR i=1 TO 4: PRINT a
$(i):"("a(i):","c(i):")":PRIN
T: NEXT i
9703 PLOT 80,70: DRAW 170,0: FOR
i=80 TO 250 STEP 20: PLOT i,71
9706 IF (i-80)/50-INT ((i-80)/50
)=0 THEN DRAW 0,4
9707 NEXT i: PRINT AT 14,9:0:AT
14,22:50
9708 PLOT 80,70: DRAW 0,102: FOR
i=70 TO 172 STEP 20: PLOT 81,i
9709 IF (i-70)/50-INT ((i-70)/50
)=0 THEN DRAW 4,0
9710 NEXT i
9720 RETURN
  
```

Con la linea 9720 il computer ritorna al programma principale per fornire le istruzioni per l'attività con carta e matita.

Attività con carta e matita

Per dare all'alunno indicazioni adatte a svolgere *liberamente e sganciato dal computer* le necessarie attività con carta e matita viene fornita una serie di opzioni prima delle domande. Lo schema è:



Opzioni

L'idea è quella di fornire direttamente all'alunno, sulla stampante, le coordinate, le istruzioni e il reticolo su cui disegnare. Se le necessità del lavoro in classe sono tali da richiedere una rapida rotazione degli alunni, ognuno potrà far conservare all'elaboratore i propri dati e cedere il posto a un compagno, altrimenti potrà restare lì ad eseguire l'attività con carta e matita e continuare l'unità con le domande.

Le opzioni possono essere interamente scavalcate premendo 0.

Questo caso si verifica sicuramente ogni volta che l'alunno, eseguita l'attività con carta e matita altrove, riprende il posto davanti al computer, richiama i dati e vuole proseguire l'unità con le domande.

Il programma delle opzioni è:

```

1703>INPUT "Ora la stampante copierà i dati che vedi scritti e ti fornirà delle istruzioni di lavoro. (Per evitare la stampa"
  
```

```

pa premi 0; altrimenti premi EN
TER);X$
1705 IF X$="0" THEN GO TO 1720
1711 FOR i=1 TO 4: LPRINT TAB 10
,a$(i);" (";a(i);" "c(i);")": LP
RINT : NEXT i: FOR i=1 TO 5: LPR
INT : NEXT i
1712 LPRINT "Ricostruisci sul re
ticolo qui disegnato il poligono
che ha i vertici nei punti di c
ui sono date le coordinate; i la
ti sono otto-nuti congiungendo o
rdinatamente i vertici."
1713 FOR i=1 TO 15: LPRINT :NEX
T i: PRINT AT 0,0; FOR i=1 TO 9
: PRINT " " : NEXT i
1714 GO SUB 6000: REM disegna re
ticolo su video e stampante
1715 CLS : GO SUB 9703: REM dise
gna su video il sist. di riferi
mento
1716 INPUT "Puoi conservare ques
ti dati se devi abbandonare il
computer e cedere il posto a un
compagno (conservare: si=s /
no=n)";S$: IF S$<>"S" AND S$<>"
n" THEN GO TO 1715
1717 IF S$="S" THEN GO TO 3000
1720 REM COMANDE
3000>REM CONSERVA I DATI E LE CO
ORDINATE

```

Disegna il reticolo

Il sistema più semplice per disegnare il reticolo è quello di farlo disegnare sul video e poi copiarlo sulla stampante. Si tratta di tracciare tante rette orizzontali quanti sono i valori possibili delle ordinate e tante rette verticali quanti sono i valori possibili delle ascisse, senza uscire dallo spazio del video e lasciando degli intervalli sufficienti a permettere la collocazione dei punti da parte dell'alunno. Si tratta inoltre di ispessire, con il tracciamento di più rette senza intervallo l'una dall'altra, il bordo di sinistra per indicare l'asse delle ordinate e quello inferiore per indicare le ascisse. Si tratta infine di tracciare dei segmenti ispessiti a intervalli di dieci spazi per fornire delle indicazioni facilitanti per gli alunni nel loro lavoro con carta e matita. La subroutine che disegna il reticolo inizia alla linea 6000 ed il suo programma è il seguente:

```

6000>REM DISEGNA IL RETICOLO
6001 CLS
6002 PLOT 1,175: DRAW 0,-174: DR
AW 254,0: PLOT 2,175: DRAW 0,-17
3: DRAW 253,0
6003 FOR i=0 TO 175 STEP 30: PLO
T 3,i+1: DRAW 2,0: PLOT 3,i+2: D
RAW 2,0: NEXT i: PLOT 3,151: DRA
W 12,0: PLOT 3,152: DRAW 12,0
6004 FOR i=0 TO 255 STEP 30: PLO
T i+1,3: DRAW 0,2: PLOT i+2,3: D
RAW 0,2: NEXT i: PLOT 151,3: DRA
W 0,12: PLOT 152,3: DRAW 0,12
6010 FOR i=0 TO 255 STEP 3: PLOT
i,0: DRAW 0,175: NEXT i
6020 FOR i=0 TO 175 STEP 3: PLOT
0,i: DRAW 255,0: NEXT i
6030 COPY
6100 LPRINT "0" 5
0"
6110 FOR i=1 TO 20: LPRINT : NEX
T i
6900 RETURN

```

Il sottoprogramma per disegnare sul video il sistema di riferimento è già stato descritto. Esso ha inizio alla linea 9700.

Nel presente utilizzo non servono però le linee da 9700 a 9702 che trasformano i valori

$x(i)$ e $y(i)$ generati RND nei valori $a(i)$ e $c(i)$ relativi alla finestra dei disegni. Questo è il motivo per cui l'istruzione che rimanda al sottoprogramma (linea 1715) non è GOSUB 9700 ma GOSUB 9703.

Con le istruzioni da 1703 a 1714 si ottiene sulla stampante, per esempio, una strisciata come quella che segue:

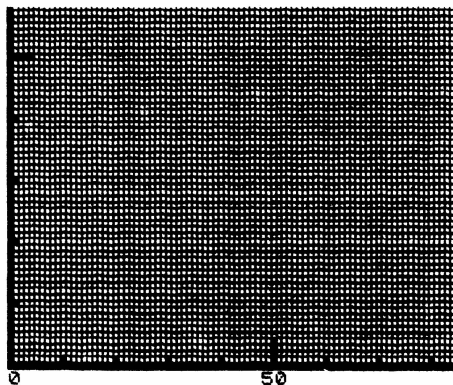
M(13,16)

N(42,20)

P(71,16)

Q(42,12)

Ricostruisci sul reticolo qui di
segnato il poligono che ha i ver-
tici nei punti di cui sono date
le coordinate; i lati sono otte-
nuti congiungendo ordinatamente
i vertici.



Disegno sul reticolo da parte dell'alunno.

Con questa attività l'alunno si esercita all'uso delle coordinate cartesiane per disegnare i quadrilateri sul reticolo fornito.

Le figure dei quadrilateri sono formate con vertici appartenenti al primo quadrante.

È una semplificazione decisa soprattutto per agevolare l'attività con carta e matita che richiede comunque una concentrazione e una precisione elevata per ragazzi di scuola media. Si è ritenuto opportuno valorizzare la fase di esercitazione sulle coordinate perché si ritiene utile agganciare il tracciamento delle figure geometriche alla loro individuazione in codice matematico.

È bene che l'alunno abbia a disposizione nella scuola media delle opportunità di astrazione (quando utilizza soltanto i valori numerici) che però non siano disgiunte da una immediata evidenziazione (quando realizza il disegno o lo osserva sul video).

Facciamo un esempio del percorso proposto per l'attività con carta e matita.

Siano stati generati i valori numerici secondo la precisione fornita nei paragrafi precedenti.

Eccone i valori casuali.

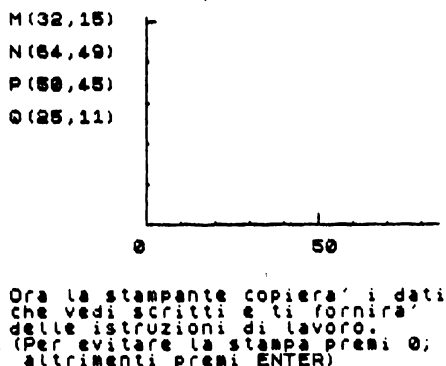
M (32,15)

N (64,49)

P (50,45)

Q (25,11)

Sul video comparirà contemporaneamente, insieme a questi valori, il disegno degli assi cartesiani e l'alternativa descritta nell'operazione (1) del diagramma di pag. 198.



Premendo 0 si scavalca l'attività con carta e matita e si passa alle domande. Premendo ENTER sulla stampante compare la strisciata prima descritta e l'alunno potrà tracciare la figura secondo le istruzioni, cioè prima individuando i punti sul piano cartesiano, poi assegnando ad ogni punto la propria lettera e successivamente unendoli secondo l'ordine delle lettere.

Intanto sul video sarà riapparso il disegno del sistema di riferimento con la frase: Puoi conservare questi dati se devi abbandonare il computer e cedere il posto a un compagno.

Rotazione degli alunni al computer per l'attività di carta e matita

L'insegnante può, volendo, far interrompere il processo durante l'attività a carta e matita, per permettere un'attività simultanea di più alunni con lo stesso elaboratore.

Questo è il motivo della richiesta in parentesi: (conservare: si = s/no = n). La risposta "s" abilita la subroutine "PER CONSERVARE I DATI E LE COORDINATE" nella quale viene chiesto all'alunno di fornire una sigla da associare alle coordinate ricevute. Questa sigla sarà la chiave attraverso cui l'alunno potrà richiamare i valori una volta eseguita l'attività con carta e matita. Il computer, ricevuta la sigla, conserva in una memoria con 9 indici (dimensionata attraverso la linea 3010) i 4 valori di x, i 4 valori di y e il valore di a che gli è indispensabile per poter "sapere" di che quadrilatero si tratta e trasferisce la memoria su microfloppy.

La lista completa della subroutine "PER CONSERVARE I DATI E LE COORDINATE" è:

```

3000>REM CONSERVA I DATI E LE CO
ORDINATE
3010 DIM L(9)
3020 FOR i=1 TO 4: LET L(i)=X(i)
: LET L(i+4)=Y(i): NEXT i: LET L
(9)=a
3030 INPUT "Scrivi una sigla e t
ienila a mente. Ti servirà per
richiamare i dati. (max. 8 caratt
eri)":V$
3040 SAVE "M":1;V$ DATA L()
3900 GO TO 1050

```

La linea 3900 rimanda al programma principale e interrompe la procedura.

A questo punto un nuovo alunno si potrà porre davanti al video e ripartire con RUN o con GOTO 1.

Si danno due casi: o il nuovo alunno sta ricominciando dall'inizio, oppure ha appena eseguito l'attività con carta e matita e deve richiamare i dati.

Il programma principale prevede queste due alternative quando, alla linea 1009, pone il quesito:

```

Stai incominciando una nuova uni
ta' oppure vuoi richiamare i da-
ti (dati=d / altrimenti ENTER)

```

Ecco le istruzioni del programma principale:

```

1009>INPUT "Stai incominciando u
na nuova unità oppure vuoi rich
iamare i dati (dati=d / altrim
enti ENTER)":S$: IF S$="d" THEN
GO SUB 4000
1010 IF S$="d" THEN GO TO 1027

```

La risposta "d" abilita la subroutine "PER RICHIAMARE I DATI E LE COORDINATE" per l'alunno che si trova nelle condizioni del secondo caso. Tale subroutine ha inizio alla riga 4000 ed eccone la lista completa:

```

4000>REM RICHIAMA I DATI E LE CO
ORDINATE
4010 INPUT "Scrivi la sigla per
richiamare i dati.":V$
4020 LOAD "M":1;V$ DATA L(): ER
ASE "M":1;V$
4030 FOR i=1 TO 4: LET X(i)=L(i)
: LET Y(i)=L(i+4): NEXT i: LET a
=L(9)
4900 RETURN

```

La linea 4900 rimanda nel programma principale alla linea 1010 che ha il compito di scavalcare la parte di programma principale con cui si fanno generare le figure.

I dati richiamati dal cartridge attraverso la sigla-chiave vengono trasformati, come è già stato spiegato nel paragrafo "Scrittura sul video delle coordinate".

Il programma completo per l'attività a carta e matita è il seguente:

```

1000>REM PROGRAMMA PRINCIPALE
1009>INPUT "Stai incominciando u
na nuova unità oppure vuoi rich
iamare i dati (dati=d / altrim
enti ENTER)":S$: IF S$="d" THEN
GO SUB 4000
1010 IF S$="d" THEN GO TO 1027
1027>GO SUB 9700: REM coordinate
e sist. di riferimento
1030 GO SUB 1700: REM opzioni
1050>CLS: INPUT "Innanzitutto,
se hai il microdrive, estrai il

```



```

cartridge. Stacca l'alimentatore
prima di spegnere il video e
la stampante "x$
1070 PRINT AT 10,11;"F I N E"
1080 GO TO 1080
1090 GO TO 1004
1099>STOP
1700>REM DOMANDE PER ATTIVITA' C
ON CARTA E MATITA
1702 LET k$=""
1703 INPUT "Ora la stampante cop
iera, i dati che vedi scritti e t
i fornira' delle istruzioni di
lavoro. (Per evitare la stam
pa premi 0; altrimenti premi EN
TER)";x$
1705 IF x$="0" THEN GO TO 1720
1711 FOR i=1 TO 4: LPRINT TAB 10

```

```

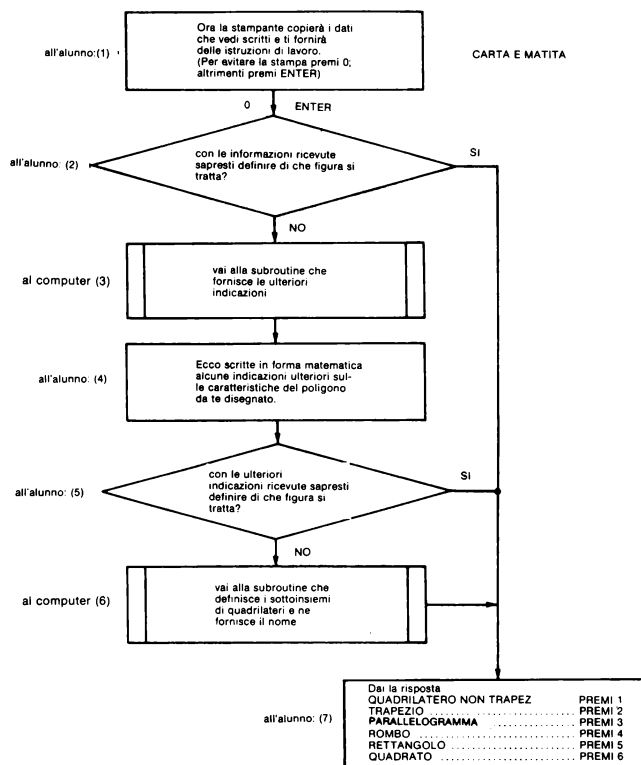
a$(i);"("a(i);"c(i);")": LP
RINT : NEXT i: FOR i=1 TO 5: LPR
INT : NEXT i
1712 LPRINT "Ricostruisci sul re
ticolo qui disegnato il poligono
che ha i vertici nei punti di c
ui sono date le coordinate; i la
ti sono otto-nuti congiungendo o
rdinatamente i vertici."
1713 FOR i=1 TO 15: LPRINT : NEX
T i: PRINT AT 0,0; : FOR i=1 TO 9
: PRINT " : NEXT i
1714 GO SUB 6000: REM disegna re
ticolo su video stampante
1715 CLS : GO SUB 9703: REM dise
gna su video il sist. di riferi
n" THEN GO TO 1715
1717 IF s$="s" THEN GO TO 3000

```

Domande

Le domande poste in sequenza di difficoltà decrescente hanno il compito di evitare, attraverso brevi salti di programma, inutili passaggi ripetitivi a chi è in grado di riconoscere la figura solo in base alle coordinate o, successivamente grazie ad ulteriori indicazioni fornite, senza ancora aver confrontato il proprio disegno a carta e matita con quello realizzato dal video. I più potranno addirittura evitare l'attività con carta e matita premendo 0.

Le domande poste in sequenza sono così organizzate:



Come si vede l'attività definita con "carta e matita" è strettamente interrelata con la II° parte, quella riferita alla domanda filtro: "che quadrilatero è?". Anzi l'operazione (7) è proprio la scrittura per la risposta a quella domanda; si rinvia alla seconda parte l'approfondimento di questo aspetto.

Le domande (2) e (5) sono il sistema per by-passare le subroutine (3) e/o (6).

L'attività con carta e matita vera e propria avviene tra l'operazione (1) e l'operazione (2).

Le ulteriori indicazioni delle subroutine (3) sono di parallelismo per il trapezio:

MN//PQ

di doppio parallelismo per i parallelogrammi:

MN//PQ
MQ//PN

di doppio parallelismo e di congruenza tra lati consecutivi per i rombi:

MN//PQ
MQ//PN
MN=NP

di doppio parallelismo e di perpendicolarità tra lati consecutivi per i rettangoli:

MN//PQ
MQ//PN
MNP:90 GRADI

di doppio parallelismo e di congruenza tra lati consecutivi e di perpendicolarità tra lati consecutivi per i quadrati:

MN//PQ
MQ//PN
MN=NP
MNP:90 GRADI

La subroutine (3) utilizza pertanto sempre la stessa stringa. Questo fatto suggerisce di utilizzare un sistema READ e DATA per conservarle e richiamarle:

```
9901>DATA "MN // PQ"
9902>DATA "MQ // PN"
9903>DATA "MN=NP"
9904>DATA "MNP:90"
```

Se a=1 il quadrilatero generato è un "non trapezio" e in questo caso non vi sono indicazioni ulteriori.

Il programma completo della subroutine che fornisce le ulteriori indicazioni è il seguente:

```
1720>REM DOMANDE
1721 INPUT "con le informazioni
ricevute sa-preti definire di c
he figura si tratta? (ENTER)
31=5 / no=0" :Z$. IF Z$<>"S" AN
D Z$<>"n" THEN GO TO 1723
1724 IF Z$="n" THEN GO SUB 1800
1725 IF Z$="S" THEN RETURN
1740 INPUT "Ecco scritte in form
a matematicaalcune indicazioni u
lteriori sulle caratteristiche d
el poligono da te disegnato.
(ENTER)";X$
```

```

1760 INPUT "con le ulteriori indicazioni ricevute sapresti definire di che figura si tratta? (ENTER sì=s / no=n)"; k$: IF k$<>"s" AND k$<>"n" THEN GOTO 1760
1790 RETURN
1800 REM INDICAZIONI ULTERIORI
1810 CLS : GO SUB 9703: PRINT AT 0,0
1820 IF a=1 THEN PRINT "NESSUN": PRINT "DATO": IF k$="s" THEN PR

```

```

INT AT 2,0:"ULTERIO-": PRINT "RE
1830 IF a=1 THEN GO TO 1840
1835 IF a>1 THEN FOR i=1 TO a-1: RESTORE 9900+i: READ b$
1834 IF a=5 AND i=3 THEN GO TO 1837
1835 IF a=6 AND i=5 THEN GO TO 1837
1836 PRINT b$: PRINT
1837 NEXT i
1840 RETURN

```

L'attività con carta e matita vera e propria avviene tra la operazione (1) e l'operazione (2).

```

1705 IF x$="0" THEN GO TO 1720
1711 FOR i=1 TO 4: LPRINT TAB 10; z$(i); "a(i)"; "c(i)"; "": LP
RINT : NEXT i: FOR i=1 TO 5: LPR
INT : NEXT i
1712 LPRINT "Ricostruisci sul re
ticolo qui disegnato il poligono
che ha i vertici nei punti di c
ui sono date le coordinate; i la
ti sono otto-nuti congiungendo o
rdinatamente i vertici."
1713 FOR i=1 TO 15: LPRINT : NEX
T i: PRINT AT 0,0: FOR i=1 TO 9
: PRINT : NEXT i
1714 GO SUB 6000
1715 CLS : GO SUB 9703
1716 INPUT "Puoi conservare ques
ti dati se devi abbandonare il
computer e cedere il posto a un
compagno (conservare: sì=s /
no=n)"; s$: IF s$<>"s" AND s$<>"
n" THEN GO TO 1715
1717 IF s$="s" THEN GO TO 3000
1720 REM DOMANDE
1723 INPUT "con le informazioni
ricevute sa-aresti definire di c
he figura si tratta? (ENTER
sì=s / no=n)"; z$: IF z$<>"s" AN
D z$<>"n" THEN GO TO 1723
1724 IF z$="n" THEN GO SUB 1800
1725 IF z$="s" THEN RETURN
1740 INPUT "Ecco scritte in form
a matematica alcune indicazioni u
lteriori sulle caratteristiche d
el poligono da te disegnato.
(ENTER)"; x$
1760 INPUT "con le ulteriori ind
icazioni ri-cevute sapresti defi
nire di che figura si tratta?
(ENTER sì=s / no=n)"; k$: IF k$<>"s" AND k$<>"n" THEN
GOTO 1760
1790 RETURN
1800 REM INDICAZIONI ULTERIORI
1810 CLS : GO SUB 9703: PRINT AT
0,0
1820 IF a=1 THEN PRINT "NESSUN":
PRINT "DATO": IF k$="s" THEN PR
INT AT 2,0:"ULTERIO-": PRINT "RE
1830 IF a=1 THEN GO TO 1840
1835 IF a>1 THEN FOR i=1 TO a-1:
RESTORE 9900+i: READ b$
1834 IF a=5 AND i=3 THEN GO TO 1
837
1835 IF a=6 AND i=5 THEN GO TO 1
837
1836 PRINT b$: PRINT
1837 NEXT i
1840 RETURN
2000 REM TEST
2070 INPUT "QUADRILATERO NON TRA
PEZ. PREMI 1TRAPEZIO-----
-----PREMI 2PARALLELOGRAMMA-----
-----PREMI 3ROMBO-----
-----PREMI 4RETTANGOLO-----
-----PREMI 5QUADRATO-----
-----PREMI 6Scrivi il numero "; b
3000 REM CONSERVA I DATI E LE CO
ORDINATE

```

```

3010 DIM l(9)
3020 FOR i=1 TO 4: LET l(i)=x(i)
: LET l(i+4)=y(i): NEXT i: LET l
(9)=a
3030 INPUT "Scrivi una sigla e t
ienila a mente. Ti servirà per
richiamare idati. (max. 8 caratt
eri)"; v$
3040 SAVE "m";1:v$ DATA l()
3900 GO TO 1060
4000 REM RICHIAMA I DATI E LE CO
ORDINATE
4010 INPUT "Scrivi la sigla per
richiamare idati."; v$
4020 LOAD "m";1:v$ DATA l(): ER
ASE "m";1:v$
4030 FOR i=1 TO 4: LET x(i)=l(i)
: LET y(i)=l(i+4): NEXT i: LET a
=l(9)
4900 RETURN
6000 REM DISEGNA IL RETICOLO
6001 CLS
6002 PLOT 1,175: DRAW 0,-174: DR
AW 254,0: PLOT 2,175: DRAW 0,-17
3: DRAW 253,0
6003 FOR i=0 TO 175 STEP 30: PLO
T 3,i+1: DRAW 2,0: PLOT 3,i+2: D
RAW 2,0: NEXT i: PLOT 3,151: DRA
W 12,0: PLOT 3,152: DRAW 12,0
6004 FOR i=0 TO 255 STEP 30: PLO
T i+1,3: DRAW 0,2: PLOT i+2,3: D
RAW 0,2: NEXT i: PLOT 151,3: DRA
W 0,12: PLOT 152,3: DRAW 0,12
6010 FOR i=0 TO 255 STEP 3: PLOT
i,0: DRAW 0,175: NEXT i
6020 FOR i=0 TO 175 STEP 3: PLOT
0,i: DRAW 255,0: NEXT i
6030 COPY
6100 LPRINT "0" 5
0"
6110 FOR i=1 TO 20: LPRINT : NEX
T i
6900 RETURN
9700 REM FORNISCE LE COORDINATE
E IL SISTEMA DI RIFERIMENTO
9701 FOR i=1 TO 4: LET a(i)=INT
((x(i)-30)/2): LET c(i)=INT ((y(
i)-70)/2): NEXT i
9702 CLS : FOR i=1 TO 4: PRINT a
$(i); " "; a(i); " "; c(i); "": PRIN
T : NEXT i
9703 PLOT 80,70: DRAW 170,0: FOR
i=80 TO 250 STEP 20: PLOT i,71
9706 IF (i-80)/50-INT ((i-80)/50
)=0 THEN DRAW 0,4
9707 NEXT i: PRINT AT 14,9:0:AT
14,22,50
9708 PLOT 80,70: DRAW 0,102: FOR
i=70 TO 172 STEP 20: PLOT 81,i
9709 IF (i-70)/50-INT ((i-70)/50
)=0 THEN DRAW 4,0
9710 NEXT i
9720 RETURN
9900 REM DATI
9900 DATA "M","N","P","O"
9901 DATA "MN // P0"
9902 DATA "M0 // PN"
9903 DATA "MN=NP"
9904 DATA "MNP:90"

```

Visualizzazione del quadrilatero sul video

Come si è visto le domande rivolte all'alunno per l'attività con carte e matita sono gradualizzate.

Ad ogni risposta negativa viene fornita un'indicazione ulteriore che avvicina al concreto il quesito posto.

Così se non sono state sufficienti le coordinate cartesiane, vengono fornite in forma matematica le condizioni di appartenenza della figura alla propria classe.

Se queste non sono sufficienti (e l'alunno non è in grado di realizzare con carta e matita la figura richiesta o non si fida della propria figura) l'elaboratore visualizza sul video il quadrilatero prima di passare alla domanda filtro: che quadrilatero è?

Il programma completo per la visualizzazione è:

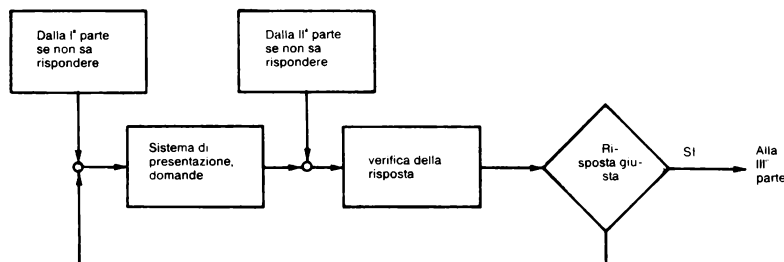
```
1000>REM PROGRAMMA PRINCIPALE
1035>IF Z$="S" OR K$="S" THEN GO
SUB 2070
1039 GO SUB 9800
1040 IF Z$="N" AND K$="N" THEN G
O SUB 2000
1041 IF Z$="N" AND K$="N" THEN G
O SUB 1800
1042 IF Z$="N" AND K$="N" THEN G
O SUB 9800: REM
9800>REM DISEGNA LA FIGURA
9810 IF S$="d" THEN GO SUB 1800
9812 PLOT X(1),Y(1): FOR I=1 TO
4: IF I=4 THEN DRAW X(1)-X(I),Y(
1)-Y(I): GO TO 9814
9813 DRAW X(I+1)-X(I),Y(I+1)-Y(I
)
9814 PRINT AT (175-Y(I))/8,X(I)/
8-1;A$(I): NEXT I
9815 PRINT AT 0,0
9820 RETURN
```

dove le linee 1035÷1041 del programma principale permettono di arrivare alla domanda filtro (linea 2070) prima della visualizzazione per chi ritenesse di aver già riconosciuto il quadrilatero.

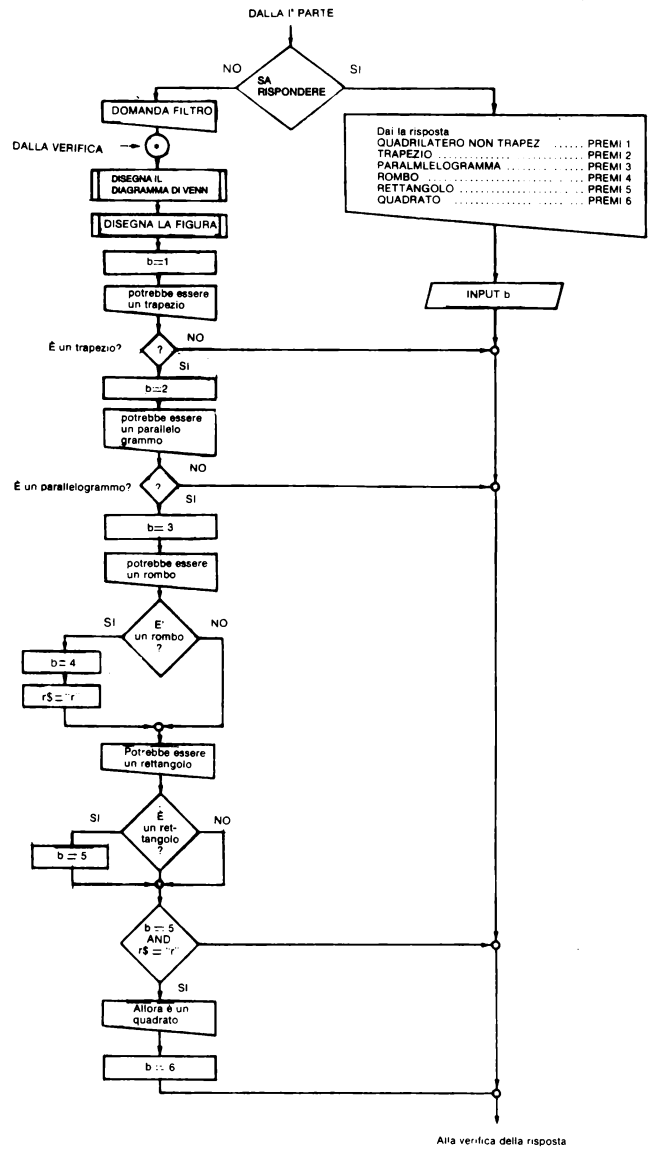
Il disegno che appare sul video è ovviamente quello già riprodotto esemplificativa-mente classe per classe.

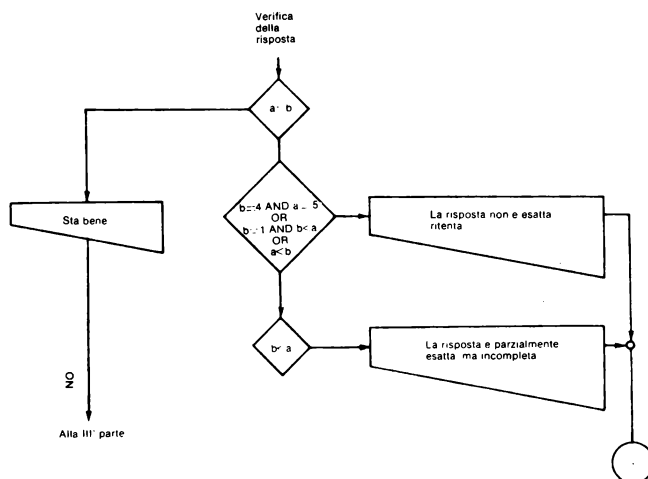
4.2.2 II^a Parte. Domanda

La domanda filtro è seguita da una serie di coppie presentazione/domanda secondo lo schema:



cui corrisponde il diagramma di flusso:





Catena di domande se l'alunno non sa rispondere

La presentazione della catena di domande (e dell'illustrazione del modo con cui i sottoinsiemi sono annodati l'uno all'altro) è opzionale perché si suppone che potrà essere necessario alla comprensione del quesito per le prime volte, ma nelle successive ripetizioni risulterà un fatto acquisito dagli alunni.

Ad essa si perviene avendo affermato, nella prima parte, di non essere in grado di riconoscere la figura.

Il programma completo per la catena di presentazioni/domande è:

```

2000>REM TEST
2010 INPUT "Osserva la figura e
i dati che sono sul video. Nota
che la figura è un quadrilater
o. DI QUALE QUADRILATER
O SI TRATTA? "
2011 INPUT "Non rispondere subito
o ma premi ENTER"
2011 GO SUB 5000
2013 GO SUB 1000
2014 GO SUB 9000
2015 INPUT "Nel seguito trovi de
lle domande. La risposta a ognuna
di queste precisa un sottoinsi
eme dell'in-sieme dei quadrilate
ri a cui appartiene questa figu
ra. Premi ENTER"
2016 LET b=1
2020>INPUT "Questa figura potreb
be essere unQUADRILATERO senza p
articolari proprietà, oppure po
trebbe averedue lati paralleli.
Premi Enter"
2021 INPUT "In quel caso diremo
che appar-tiene a quel sottoin
sieme di quadrilateri che si chi
amano
DOMANDA: questa figu
ra è un tra-pezio? (s/n)"
2021 IF x$="n" AND x$<>"s" THEN GO TO
2021
2022 IF x$="n" THEN GO TO 2072
2030 LET b=2: INPUT "Può essere
un TRAPEZIO qualun-que, oppure
avere paralleli an-che gli alt
ri due lati. Premi
  
```

Subroutine che disegna il dia-
gramma di Venn

Serie di subroutines che diseg-
nano la figura, già illustrata
nella II° parte

```

ENTER";X$
2031>INPUT "In quel caso diremmo
che appar- tiene a quel sottoin-
sieme di quadrilateri che si chi-
amano < PARALLELOGRAMMI
>". DOMANDA: questa figu-
ra e' un pa- rallelogramma? (s/n)
";X$; IF X$<>"n" AND X$<>"s" THE
N GO TO 2031
2032 IF X$="n" THEN GO TO 2072
2040 LET b=3: INPUT "Puo' essere
un PARALLELOGRAMMA qualsiasi,
senza altre connota- zioni. Oppu-
ra avere i lati tra loro congru-
enti. Premi ENTER";X$
2041 LET r$="": INPUT "In quel c-
aso diremmo che la figura e' un
< ROMBO >". DOMANDA:
questa figura e' un rombo? (s/n)
";X$; IF X$<>"s" AND X$<>"n" TH
EN GO TO 2041
2042>IF X$="s" THEN LET b=4: LET
r$="r"
2045 INPUT "Oppure puo' essere u-
n PARALLELO-GRAMMA che ha gli a-
ngoli con- gruenti tra di loro.
Premi ENTER";X$
2046 INPUT "In questo caso dicia-
mo che la figura e' un < RETTAN-
GOLO >". DOMANDA: questa figu-
ra e' un rettangolo? (s/n);X$;
IF X$<>"s" AND X$<>"n" THEN GO T
O 2046
2047 IF X$="s" THEN LET b=5
2048 IF b=5 AND r$="r" THEN LET
b=6: INPUT "Allora questa figura
e' un < QUADRATO >".";X$
GO TO 2072
2065 GO TO 2072

```

La linea 2065 permette di scavalcare la parte di programma a cui è rinviato l'alunno che afferma di saper riconoscere la figura.

Sul video apparirà questo testo:

```

Osserva la figura e i dati che
sono sul video. Nota che la figu-
ra e' un quadrilatero.
DI QUALE QUADRILATERO SI TRATTA?
Non rispondere subito ma premi
ENTER

```

```

Nel seguito trovi delle domande.
La risposta a ognuna di queste
precisa un sottoinsieme dell'in-
sieme dei quadrilateri a cui ap-
partiene questa figura.
Premi ENTER

```

```

Questa figura potrebbe essere un
QUADRILATERO senza particolari p-
roprietà, oppure potrebbe avere
due lati paralleli. Premi
Enter

```

```

In quel caso diremmo che appar-
tiene a quel sottoinsieme di qua-
drilateri che si chiamano
< TRAPEZI >".
DOMANDA: questa figura e' un tra-
pezio? (s/n)

```

```

Puo' essere un TRAPEZIO qualun-
que, oppure avere paralleli an-
che gli altri due lati.
Premi ENTER

```

In quel caso diremmo che appartiene a quel sottoinsieme di quadrilateri che si chiamano < PARALLELOGRAMMI >.
 DOMANDA: questa figura e' un parallelogramma? (s/n)

Puo' essere un PARALLELOGRAMMA qualsiasi, senza altre connotazioni. Oppure avere i lati tra loro congruenti.
 Premi ENTER

In quel caso diremmo che la figura e' un < ROMBO >.
 DOMANDA: questa figura e' un rombo? (s/n)

Oppure puo' essere un PARALLELOGRAMMA che ha gli angoli congruenti tra di loro.
 Premi ENTER

In questo caso diciamo che la figura e' un < RETTANGOLO >.
 DOMANDA: questa figura e' un rettangolo? (s/n)

Per ogni domanda è prevista la risposta dell'alunno e si può dare il caso che sia stata data risposta affermativa agli ultimi due quesiti. In quel caso comparirà il testo:

Allora questa figura e' un < QUADRATO >.

Se l'alunno sa rispondere

L'alunno perviene a questa parte di programma avendo affermato nella prima parte di essere in grado di riconoscere il quadrilatero. Si osservi che in questo caso l'alunno non ha visto il disegno della figura se non per averlo eventualmente disegnato con carta e matita.

Il test che viene posto è diretto, cioè scavalca la catena di domande, e il programma è:

```
2070>INPUT "QUADRILATERO NON TRA
PEZ. PREMI 1TRAPEZIO-----
---->PREMI 2PARALLELOGRAMMA-----
---->PREMI 3ROMBO-----
---->PREMI 4RETTANGOLO-----
---->PREMI 5QUADRATO-----
---->PREMI 6Scrivi il numero ";b
2072 IF b<1 OR b>6 THEN GO TO 20
70
```

dove la linea 2072 evita che si diano errori di tastiera.

Verifica della risposta

Tale verifica è imposta sul confronto tra le due variabili: a (valore generato RND cui corrisponde la classe del quadrilatero generato dal computer) e b (definita dall'utente o direttamente o attraverso la catena di domande).

Si danno i tre casi generali e due casi particolari:

casi generali:

I° $a=b$ Allora l'alunno ha risposto esattamente e si passa alla III° parte

II° $a<b$ Allora l'alunno ha collocato la figura in un sottoinsieme a cui essa non appartiene, poiché essa appartiene a un insieme o sottoinsieme più generale.

La risposta implica l'attribuzione alla figura di proprietà che non possiede dunque è errata.

III° $b < a$ Allora l'alunno ha collocato la figura in un insieme o in un sottoinsieme a cui essa appartiene. La risposta implica non aver preso in considerazione tutte le proprietà definite per la figura. Dunque la risposta è esatta, ma incompleta.

casi particolari:

I° $b=1$ e $b < a$ Allora l'alunno avrebbe definito quadrilatero non trapezio un quadrilatero trapezio. In questo caso la risposta è errata.

II° $b=4$ e $a=5$ Allora l'alunno avrebbe definito rombo una figura che è un rettangolo. Anche in questo caso la risposta è errata e non solo incompleta.

Il programma per la verifica della risposta è:

```
2010>IF a=b THEN GO TO 2250
2015 IF b=1 AND b<a OR b=4 AND a
2020 THEN INPUT "La risposta non è
    esatta: ritenta." (P
    remi ENTER) "X$": GO TO 2011
2020 IF b<a THEN INPUT "La rispo
    sta è parzialmente esatta, ma i
    ncompleta." (ENTER)
    "X$": GO TO 2011
2030 IF a<b THEN INPUT "La rispo
    sta non è esatta: ritenta."
    (ENTER)
    "X$": GO TO 2011
2250 PRINT FLASH 1; AT 19,0: "STA
    BENE." PAUSE 100: PRINT AT 19,0
2290 RETURN
```

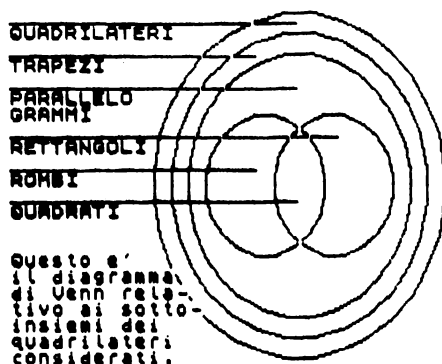
Diagramma di Venn dei quadrilateri

Il diagramma di Venn è uno strumento visivo per la classificazione.

Esso rappresenta il modo in cui un insieme di oggetti trova una relazione con un altro. I modi possibili sono che un insieme sia incluso nell'altro o che due insiemi si intersechino o che siano uno esterno all'altro.

Gli oggetti in questo caso sono quadrilateri. Sono state prese in considerazione alcune classi, le più generali, e sono stati trascurati i casi particolari come è stato già specificato nella I° parte parlando della variabile a.

Si è cioè usato il diagramma dei quadrilateri come viene rappresentato nella maggioranza dei testi di matematica per la scuola media.



Il sottoprogramma completo è:

```

5000>REM DIAGRAMMA DI VENN
5010 CLS
5100 OVER 1
5200 CIRCLE 168,88,85
5202 PRINT "_____
5204 PRINT "QUADRILATERI": PAUSE
5206 PRINT "_____
5210 CIRCLE 168,88,75
5212 PRINT "_____
5214 PRINT "TRAPEZI": PAUSE 20
5220 CIRCLE 168,88,65
5222 PRINT "_____
5224 PRINT "PARALLELO"
5226 PRINT "GRAMMI": PAUSE 20
5240 CIRCLE 168,88,35
5242 PRINT "_____
5244 PRINT "RETTANGOLI": PAUSE 2
5246 PRINT "_____
5250 CIRCLE 148,88,35
5400 PRINT "_____
5410 PRINT "ROMBI": PAUSE 40
5420 PRINT "_____
5430 PRINT "QUADRATI": PAUSE 20
5500 OVER 0
5510 PRINT : PRINT
5520 PRINT "Questo è": PRINT "i
l diagramma": PRINT "di Venn rel
a": PRINT "tivo ai sotto-": PRI
NT "insiemi de
5530 PRINT "quadrilateri": PRIN
T "considerati."
5500 INPUT "Premi ENTER";x$
5900 RETURN

```

Il sottoprogramma per disegnare la figura viene eseguito attraverso le linee 2012, 2013, 2014.

Esso risulta necessario per far riapparire la figura dopo la rappresentazione del diagramma di Venn.

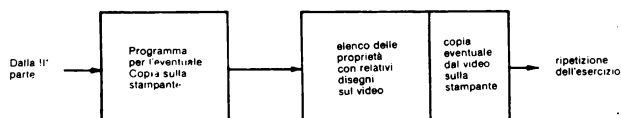
Ovviamente nel rimandare alla subroutine si saltano le prime linee (9700÷9704) che servono a trasformare le coordinate sul video in coordinate della finestra. Si rimanda alla prima parte per la listatura.

4.2.3 III^a Parte. Descrizione

La terza parte ha la funzione di far osservare un certo corredo di proprietà di cui è in possesso la figura rappresentata e di far osservare come alcune di queste sono proprie del sottoinsieme a cui la figura appartiene, mentre altre sono proprie dell'insieme più ampio a cui il sottoinsieme, a sua volta, appartiene.

L'alunno si potrà così esercitare ad osservare la modalità con cui una proprietà di un insieme superiore si traduce nel sottoinsieme a cui appartiene la figura.

Lo schema della III^o parte è:

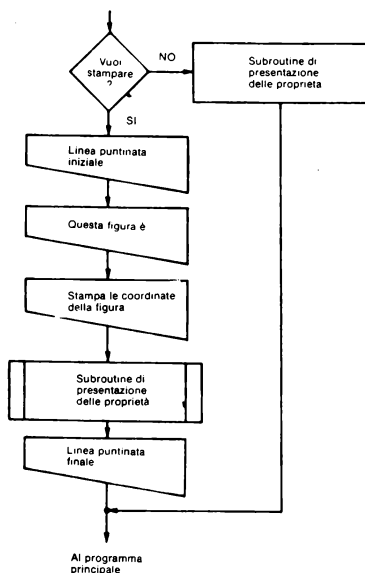


Copia sulla stampante

È previsto che l'alunno possa conservare, per esempio su un quaderno, a fianco del proprio disegno sul reticolo, il disegno eseguito dall'elaboratore corredato dall'indicazione delle proprietà e dei diagrammi forniti.

Il programma che riguarda la copia su stampante definisce con la linea puntinata, l'inizio della stampa per la presente figura in modo da isolarla da eventuali altre parti a stampa precedenti. Definisce con lo stesso sistema la fine della stampa. Fornisce inoltre le coordinate della figura.

Il diagramma è il seguente:



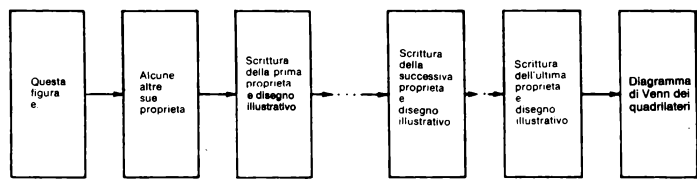
Il programma completo è:

```

8000>REM STAMPA
8002 INPUT "vuoi stampare (s/n)
";ys: IF ys<>"s" AND ys<>"n" THE
N GO TO 8002
8004 IF ys="n" THEN GO SUB 7000
8006 IF ys="s" THEN GO TO 8013
8010 RETURN
8013 LPRINT : LPRINT "....."
      TO 10: LPRINT : NEXT i
8015 FOR i=1 TO 4: LPRINT a$(i);
      "("a$(i);";";c(i);")": NEXT i: L
PRINT
8020 COPY : PRINT AT 15,0:"QUEST
A FIGURA E' ": LPRINT "QUESTA FI
GURA E'
8030 RESTORE 9905: FOR i=1 TO a:
READ d$: NEXT i: LPRINT : LPRIN
T d$: PRINT d$
8031 FOR i=1 TO 4: LPRINT : NEXT
i: LPRINT : LPRINT "ALCUNE SUE
PROPRIETA'
8032 GO SUB 7000
8034 FOR i=1 TO 10: LPRINT : NEX
T i: LPRINT "....."
8035 FOR i=1 TO 30: LPRINT : NEX
T i
8040 RETURN
  
```

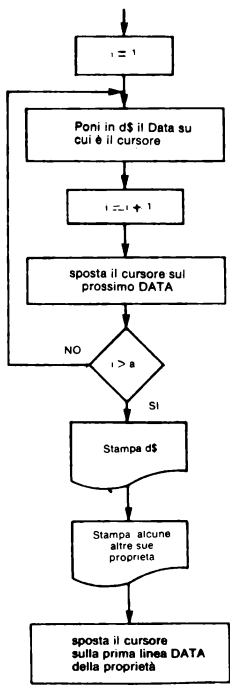
Elenco delle proprietà

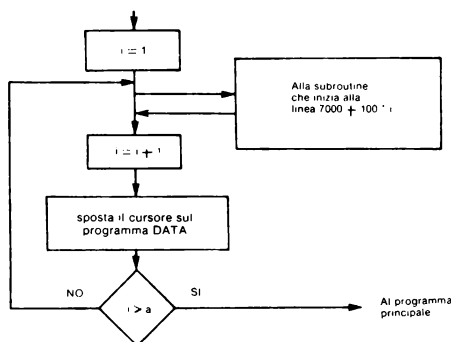
Il programma che evidenzia sul video le proprietà del quadrilatero di cui sono state fornite le coordinate è organizzato secondo lo schema:



Il nome della figura viene scelto dal computer utilizzando la variabile *a* e le istruzioni di READ e DATA. Si tratta di far scorrere il cursore dell'istruzione READ su una linea di DATA sulla quale i nomi dei sottoinsiemi considerati siano stati inseriti rispettando l'ordine con cui sono state generate le classi di figure. Il cursore deve spostarsi tante volte quanto vale il numero contenuto nella memoria *a* e ogni volta inserire nella memoria *d\$* il nome che legge.

Alla fine stamperà l'ultimo nome letto che corrisponderà alla figura generata. Il diagramma di flusso è il seguente:





In questo diagramma sono state omesse, per semplicità, le operazioni che riguardano la stampa su carta delle proprietà evidenziate sul monitor.

Il programma risulta perciò:

```

7000 REM VISUALIZZA SUL VIDEO LE
7001 PROPRIETA
7020 PRINT AT 15,0;"QUESTA FIGUR
7021 A E"
7030 RESTORE 9905: FOR i=1 TO a:
7031 READ d$: NEXT i: PRINT d$
7031 PRINT AT 19,0;"ALCUNE ALTRE
7032 SUE PROPRIETA"
7032 FOR i=1 TO a: INPUT "premi
7033 ENTER";x$: CLS: RESTORE 9905+i
7034 READ e$: PRINT AT 4,5,i;" ";e
7035 GO SUB 7000+100*i
7036 IF y$="s" THEN FOR v=1 TO 4
7037 : LPRINT: NEXT v
7038 NEXT i
7039 INPUT "premi ENTER";x$
7040 CLS: IF y$="s" THEN GO SUB
7041 5000: COPY
7042 IF y$="s" THEN FOR v=1 TO 4
7043 : LPRINT: NEXT v
7044 RETURN
  
```

Analizziamo ora le singole proprietà.

Proprietà dei quadrilateri

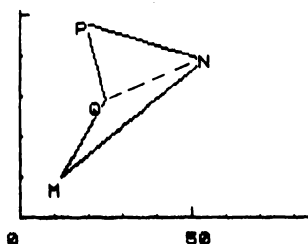
Tutti i quadrilateri godono della proprietà generale per cui, se non sono intrecciati, cioè se non hanno lati che si intersecano, la somma degli angoli interni, calcolata in gradi, è 360.

Questa proprietà vale tanto per i quadrilateri convessi che per quelli concavi. Infatti nel caso che segue, per esempio:

```

M(12,10)
N(54,39)
P(19,48)
O(25,29)
  
```

NESSUN
DATO



è evidente che:

$$\widehat{MQP} = \widehat{MQN} + \widehat{NQP}$$

e che:

$$\widehat{MNP} = \widehat{MNQ} + \widehat{QNP}$$

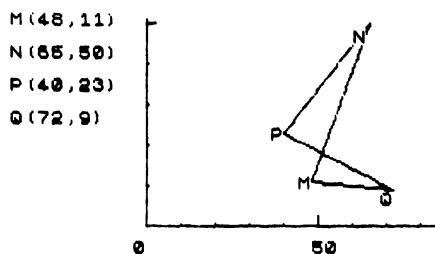
La somma degli angoli interni del triangolo QMN è 180.

La somma degli angoli interni del triangolo PNQ è 180.

MNPQ ha dunque la somma degli angoli interni di 360.

Questa proprietà non vale per i quadrilateri intrecciati.

Infatti nel caso che segue, per esempio:



i quattro angoli del quadrilatero sono \widehat{MPP} , \widehat{NPQ} , \widehat{PQM} , \widehat{QMN} ed è evidente che essendo tutti e quattro gli angoli acuti (< 90) hanno una somma che risulta inferiore a 360.

Questa particolarità viene evidenziata nell'elenco delle proprietà per il sottoinsieme dei quadrilateri non trapezi.

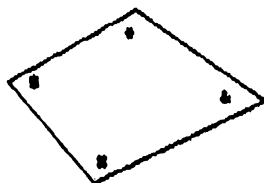
Ecco come appare la lista delle proprietà per tale sottoinsieme:

QUESTA FIGURA E'

UN QUADRILATERO NON TRAPEZIO

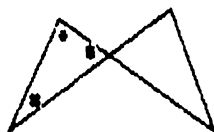
ALCUNE SUE PROPRIETA'

1. I quadrilateri non intrecciati hanno la somma degli angoli interni di 360.



$$a + b + c + d = 360$$

1 bis. La somma degli angoli interni di un quadrilatero intrecciato non è 360 in gradi perché i due angoli opposti che hanno come vertice l'intersezione tra due lati non sono angoli del quadrilatero.



$a + b + c + d = 180$
 s non è un angolo del quadrilatero

QUADRILATERI

TRAPEZI

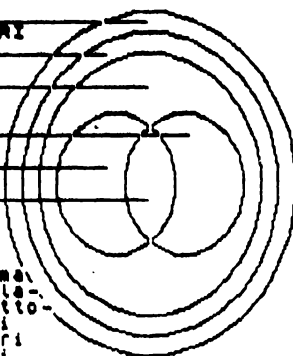
PARALLELOGRAMMI

RETTANGOLI

ROMBI

QUADRATI

Questo è il diagramma di Venn relativo ai sotto-insiemi dei quadrilateri considerati.



Il programma completo per le proprietà dei quadrilateri, è:

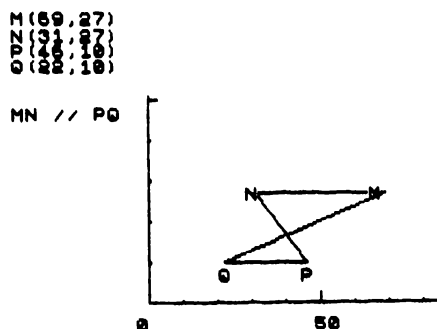
```
7100>REM PROPRIETA' DI QUADRILATE
R1
7110 PLOT 95,9: DRAW 100,40: DRA
W -75,45: DRAW -75,-35: DRAW 50,
-50
7120 PRINT AT 21-(17/8),95/8;"#"
: PRINT AT 21-(49/8),170/8;"&"
PRINT AT 21-(80/8),110/8;"$": PR
INT AT 21-(56/8),55/8;"s"
7125 IF y$="s" THEN COPY: LPRIN
T "s + & + * + $ = 360"
7127 INPUT "s + & + * + $ = 360
(ENTER)";x$
7181 CLS: PRINT AT 2,5;"1 bis.
La somma degli angoli interni d
i un quadrilatero intrecciato no
n è 360 in gradi perché i du
e angoli opposti che hanno come v
ertice l'intersezione tra due l
ati non sono angoli del quadrila
7182>PLOT 60,28: DRAW 95,60: DRA
W -25,-60: DRAW -90,55: DRAW -30,
-55
7185 PRINT AT 21-30/8-1,70/8;"#"
: PRINT AT 21-78/8+1,95/8-1;"s"
: PRINT AT 21-66/8,100/8;"$": PRI
NT AT 19,0;"s + * + $ = 180"
7186 PRINT "s non è un angolo d
el quadrilatero": IF y$="s" THE
N COPY
7190 RETURN
9906>DATA "I quadrilateri non
intrecciati hanno la somma d
e gli angoli interni di 360."
```

Proprietà dei trapezi

Per i trapezi è stata adottata la definizione più generale presente nei testi scolastici: sono quei quadrilateri che hanno due lati paralleli.

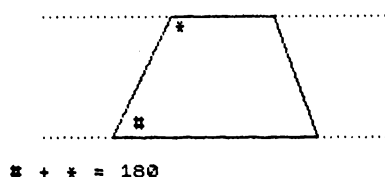
Si intende così che sono trapezi tutti quei quadrilateri che hanno almeno due lati paralleli: anche i quadrati!

Da questa definizione così ampia e non limitativa deriva che non si può escludere che nel sottoinsieme dei trapezi siano presenti dei quadrilateri intrecciati come nel caso che segue.

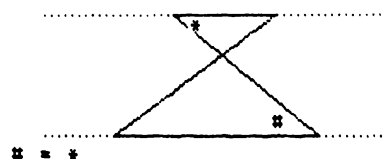


Valgono per i trapezi tutte le affermazioni fatte per i quadrilateri e inoltre questa altra proprietà.

2. I trapezi non intrecciati hanno $\alpha + \beta = 180$. Infatti si tratta di angoli coniugati formati da due parallele tagliate da una trasversale.



2 bis. In un trapezio intrecciato $\alpha = \beta$ perché angoli alterni interni formati da rette parallele tagliate da una trasversale.



Il programma completo per le proprietà dei trapezi è:


```

7200>REM PROPRIETA' DEI TRAPEZI
7202 FOR h=20 TO 220 STEP 4: PLO
T h,12: PLOT h,72: NEXT h
7210 PLOT 60,12: DRAW 120,0: DRA
U -25,60: DRAW -60,0: DRAW -35,-
60
7220 PRINT AT 21-16/8,70/8;"#":
PRINT AT 21-62/8,95/8;"*":
7230 IF y$="s" THEN COPY : LPRIN
T "# + * = 180"
7232 INPUT "# + * = 180      ENT
ER";x$
7240 CLS : PRINT AT 5,5;"2 bis.
In un trapezio in-   trecciato #
= * perche' angoli alterni inte
rni formati da ret- te parallele
tagliate da una   trasversale.

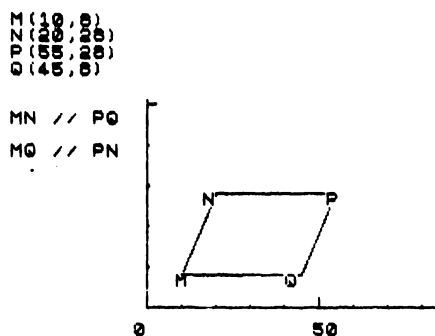
7242>FOR h=20 TO 220 STEP 4: PLO
T h,12: PLOT h,72: NEXT h
7244 PLOT 60,12: DRAW 120,0: DRA
U -85,60: DRAW 60,0: DRAW -95,-6
0
7248 PRINT AT 21-16/8,165/8-2;"#
": PRINT AT 21-62/8,95/8+1,"*":
PRINT AT 21,0,"# = *":
7250 IF y$="s" THEN COPY
7290 RETURN
9890>REM DATI
9907>DATA "I trapezi non intrecc
ia-ti hanno * + # = 180. Infatti
si tratta di angoli coniugati
formati da due parallele tagli
ate da una trasversale."

```

Proprietà dei parallelogrammi

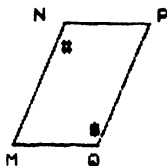
Per i parallelogrammi la definizione di riferimento è quella restrittiva rispetto ai trapezi: sono quei trapezi che hanno i lati opposti paralleli. Si intende così che sono parallelogrammi i quadrilateri che hanno i quattro lati a due a due paralleli.

Da questa definizione deriva l'impossibilità di parallelogrammi intrecciati. Ecco un esempio:

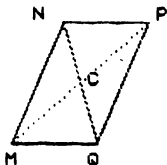


Valgono per i parallelogrammi le proprietà dei trapezi e inoltre le seguenti:

3. Tutti i parallelogrammi hanno gli angoli opposti congruenti, i lati opposti congruenti. Le diagonali si intersecano nel punto medio.



$MN = PQ$
 $\# = S$



$MC = CP$
 $NC = CQ$

Il programma completo per le proprietà dei parallelogrammi è:

```

7300>REM PROPRIETA' DEI PARALLEL
OGRAMMI
7310 DIM s(4): DIM t(4): LET s(1)
)=68: LET s(2)=s(1)+30: LET s(3)
)=148: LET s(4)=s(3)-30: LET t(4)
)=19: LET t(1)=t(4): LET t(3)=79:
LET t(2)=t(3)
7320 PLOT s(1),t(1): FOR g=1 TO
4: IF g=4 THEN DRAW s(1)-s(g),t(
1)-t(g): GO TO 7340
7330 DRAW s(g+1)-s(g),t(g+1)-t(g)
7340 IF g=1 OR g=4 THEN PRINT AT
(175-t(g))/8,s(g)/8-1;as(g): GO
TO 7350
7345 IF g=2 THEN PRINT AT (175-t
(g))/8-1,s(g)/8-2;as(g): GO TO 7
350
7347 PRINT AT (175-t(g))/8-1,s(g
)/8;as(g)
7350 NEXT g
7370>INPUT "MN = PQ" pre
mi ENTER:xs
7372 PRINT AT (175-t(2))/8+1,s(1
)/8+3;"#": PRINT AT (175-t(4))/8
-2,s(4)/8-1;"S"
7375 IF v$="s" THEN COPY : LPRIN
T "MN = PQ": LPRINT "S = S"
7380 INPUT "S = S" pre
mi ENTER:xs
7381 PRINT AT (175-t(2))/8+1,s(1
)/8+3;"#": PRINT AT (175-t(4))/8
-2,s(4)/8-1;"S"
7382 FOR g=1 TO 80 STEP 3: PLOT
s(1)+g,t(1)+3/4+g: NEXT g
7384 FOR g=1 TO 4: s(4)-s(2) STEP
5: PLOT s(2)+g,t(2)-3+g: NEXT g
7385 PRINT AT (175-(t(1)+40))/8,
(s(1)+40)/8;"C"
7387 PRINT AT 15,0;"MC = CP": PR
INT AT 15,0;"NC = CQ"
7388 IF v$="s" THEN FOR v=1 TO 1
4: PRINT " ": NEXT v: COPY
7390>RETURN
9900>DATA "Tutti i parallelogram
- mi hanno gli angoli opposti c
on-gruenti, i lati opposti congr
uenti. Le diagonali si intersec
ano nel punto medio."

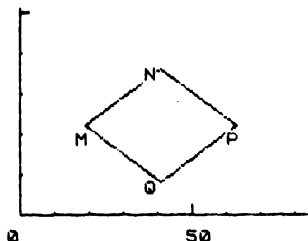
```

Proprietà dei rombi

Per i rombi la definizione di riferimento può essere indifferentemente restrittiva rispetto ai parallelogrammi o ai quadrilateri: sono quei parallelogrammi che hanno i lati tra loro congruenti. Infatti non si dà quadrilatero con i lati congruenti che non sia anche parallelogramma. Ecco un esempio:

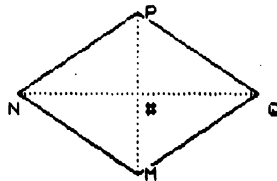
```
M(19,22)
N(41,36)
P(63,22)
Q(41,8)
```

```
MN // PQ
MQ // PN
MN=NP
```

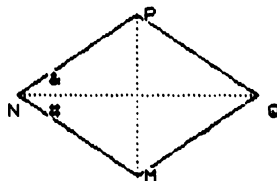


Valgono per i rombi le proprietà dei parallelogrammi e inoltre le seguenti:

4. Tutti i rombi hanno le diagonali perpendicolari. Ogni diagonale è bisettrice di due angoli opposti.



$\angle = 90$



$\angle = \angle$

Il programma completo per le proprietà dei rombi è:

```
7400>REM PROPRIETA' DEI ROMBI
7410 LET s(1)=100: LET s(3)=s(1)
: LET s(2)=30: LET s(4)=s(2)+140
: LET t(1)=11: LET t(3)=t(1)+60
: LET t(2)=t(1)+40: LET t(4)=t(2)
```

```

7420 PLOT s(1),t(1): FOR g=1 TO
4: IF g=4 THEN DRAW s(1)-s(g),t(
1)-t(g): GO TO 7435
7430 DRAW s(g+1)-s(g),t(g+1)-t(g
)
7435 NEXT g
7440 PRINT AT (175-t(1))/8-1,s(1
)/8;"s(1)";
7442 PRINT AT (175-t(2))/8,s(2)/
8-1;"s(2)";
7443 PRINT AT (175-t(3))/8-1,s(3
)/8;"s(3)";
7445 PRINT AT (175-t(4))/8,s(4)/
8+1;"s(4)";
7460>FOR g=1 TO 140 STEP 3: PLOT
s(2)+g,t(2): NEXT g: FOR g=1 TO
80 STEP 3: PLOT s(1),t(1)+g: NE
XT g
7461 PRINT AT (175-(t(1)+40))/8,
s(1)/8;"#";
7462 IF v$="s" THEN COPY : LPRIN
T "# = 90"
7465 INPUT "# = 90      Premi E
NTER";x$
7470 PRINT AT (175-(t(1)+40))/8,
s(1)/8;" ": PRINT AT 21,0;"#";
7472 PRINT AT (175-t(2))/8,s(2)/
8+2;"#"; PRINT AT (175-t(2))/8-2
,s(2)/8+2;"&": PRINT AT 21,0;"#";
7480 IF v$="s" THEN FOR v=1 TO 7
: PRINT " ": NEXT v: COPY
7490 RETURN
9909>DATA "Tutti i rombi hanno l
e diagonali perpendicolari. Ogn
i diagonale e' bisettrice di du
e angoli opposti."

```

Proprietà dei rettangoli

Per i rettangolo la definizione è riferita ai parallelogrammi: sono rettangoli quei parallelogrammi che hanno un angolo retto.

Infatti basta che un angolo di un parallelogramma sia retto perché anche gli altri lo siano.

N.B. Non è opportuno fidarsi della sola figura che la stampante realizza su carta in quanto molte stampanti producono alterazioni del disegno; vanno invece considerate le coordinate assegnate e scritte su video.

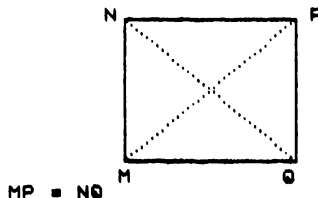
$\hat{NM}Q$ è retto in quanto supplementare di un angolo retto (\hat{MNP}).

Infatti NM è una trasversale tra le parallele $MQ//PN$ (vedi i dati). Si è applicata al rettangolo una proprietà di tutti i trapezi.

$\hat{MQ}^{\wedge}P$ è retto per essere opposto a un angolo retto (\hat{MNP}) in un parallelogramma.

Un rettangolo è un parallelogramma e valgono tutte le proprietà dei parallelogrammi cui si aggiunge:

5. Tutti i rettangoli hanno le diagonali congruenti.



Il programma completo delle proprietà dei rettangoli è:

```

7500>REM PROPRIETA' DEI RETTANGO
L1
7510 LET s(1)=68: LET s(2)=s(1):
LET s(3)=168: LET s(4)=s(3): LE
T t(4)=19: LET t(1)=t(4): LET t(
3)=89: LET t(2)=t(3)
7520 PLOT s(1),t(1): FOR g=1 TO
4: IF g=4 THEN DRAW s(1)-s(g),t(
1)-t(g): GO TO 7540
7530 DRAW s(g+1)-s(g),t(g+1)-t(g
)
7540 IF g=1 OR g=4 THEN PRINT AT
(175-t(g))/8,s(g)/8-1;a$(g): GO
TO 7550
7545 IF g=2 THEN PRINT AT (175-t
(g))/8-1,s(g)/8-2;a$(g): GO TO 7
550
7547 PRINT AT (175-t(g))/8-1,s(g
)/8+1;a$(g)
7550 NEXT g
7560>FOR g=1 TO 100 STEP 3: PLOT
s(1)+g,t(1)+7/10*g: PLOT s(2)+g
,t(2)-7/10*g: NEXT g
7565 PRINT AT 21,0:"MP = NO"
7570 IF y$="s" THEN COPY
7590 RETURN
9910>DATA "Tutti i rettangoli ha
nnole diagonali congruenti."

```

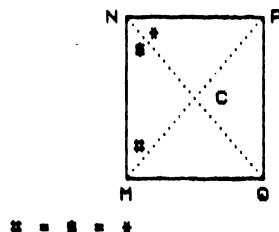
Proprietà dei quadrati

Per i quadrati esistono molte definizioni analoghe come significato e conseguenze. Essi sono infatti rombi con le diagonali congruenti, rettangoli con i lati congruenti, rettangoli con le diagonali perpendicolari, rombi con un angolo retto.

Tutte queste definizioni esplicitano in un modo o nell'altro l'appartenenza della figura insieme alle due classi dei rettangoli e dei rombi.

Occorre osservare che la stampante non restituisce un'immagine esatta del quadrato se, come nella SEIKOSHA GP 50S, il pixel di stampa occupa uno spazio più ampio in verticale che in orizzontale. Ecco l'osservazione conclusiva che viene presentata all'allunno.

5. Un quadrato e' infatti
sia un rombo che un rettangolo.



Il programma completo riferito alla conclusione sui quadrati è:

```

7600>REM PROPRIETA' DEI QUADRATI
7610 LET s(1)=68: LET s(2)=s(1):
LET s(3)=148: LET s(4)=s(3): LE
T t(4)=27: LET t(1)=t(4): LET t(
3)=107: LET t(2)=t(3)
7620 PLOT s(1),t(1): FOR i=1 TO
4: IF i=4 THEN DRAW s(1)-s(i),t(
1)-t(i): GO TO 7640

```

```

7630 DRAW s(i+1)-s(i),t(i+1)-t(i)
)
7640 IF i=1 OR i=4 THEN PRINT AT
(175-t(i))/8,s(i)/8-1;as(i): GO
TO 7650
7645 IF i=2 THEN PRINT AT (175-t
(i))/8-1,s(i)/8-2;as(i): GO TO 7
650
7647 PRINT AT (175-t(i))/8-1,s(i)
)/8;as(i)
7650 NEXT i
7660>FOR i=1 TO 80 STEP 3: PLOT
s(1)+i,t(1)+i: PLOT s(2)+i,t(2)-
i: NEXT i
7665 PRINT AT (175-(t(1)+45))/8,
(s(1)+55)/8;"C"
7667 PRINT AT (175-t(1))/8-3,s(1)
)/8;"$": PRINT AT (175-t(2))/8+1
,s(2)/8;"$": PRINT AT (175-t(2))
/8,s(2)/8+1;"*"
7670 PRINT AT 21,0;"$ = $ = *"
7680 IF ys="s" THEN COPY
7900 RETURN
9911>DATA "Un quadrato e' infatt
i sia un rombo che un rettangol
o."

```

Per ripetere l'esercizio

La parte di iterazione dell'esercizio è inserita nel programma principale.
Di essa si dà solo la listatura:

```

1044>REM PROCEDURA DI ITERAZIONE
DEL PROGRAMMA
1045 CLS : PRINT AT 10,0;"Vuoi r
iprovare? (s=s / no=n)"
1046 IF INKEY$<>" " THEN GO TO 10
46
1047 IF INKEY$<>"s" AND INKEY$<>
"n" THEN GO TO 1047
1048 IF INKEY$=" " THEN GO TO 104
8
1049 IF INKEY$<>"n" THEN GO TO 1
090
1050 CLS : INPUT "Innanzitutto,
se hai il microdrive, estrai il
cartridge. Stacca l'alimentatore
prima di spegnere il video e
la stampante.";X$
1070 PRINT AT 10,11;"F I N E"
1080 GO TO 1080
1090 GO TO 1004
1099 STOP

```

4.3 I Giudizi del Parlante

LUIGI D'ANGELO, LUCIA LEOMBRUNO, LIVIO MANDELLI

Premessa

Uno degli obiettivi che la scuola oggi si propone è quello di un "sapere articolato ed insieme unitario". Per ottenere ciò è necessario evidentemente superare ogni tipo di specializzazione e divisione di campi di intervento. Muovendosi in questa direzione anche l'educazione linguistica è andata perdendo la sua specificità e, nella sua più ampia accezione, è vista oggi come occasione per consentire l'acquisizione della più vasta gamma di linguaggi, da quello più specificatamente letterario a quello tecnico, a quello scientifico, ecc.

L'ingresso del computer nel mondo della scuola non deve a nostro avviso, andare solo a vantaggio di alcune aree disciplinari. Come insegnanti di lettere, sulla base delle conoscenze da noi acquisite in questi ultimi ci siamo posti il problema di studiare come utilizzare il computer in campo linguistico ed, in particolare, nello studio della "riflessione sul linguaggio".

Un approccio razionale allo studio della lingua presuppone non solo un ripensamento sui contenuti e sulle metodologie, ma anche una ricerca di nuovi strumenti didattici. Nella elaborazione della nostra ipotesi di lavoro ci siamo preoccupati di scegliere un modello linguistico cui fare riferimento e di individuare un momento dell'attività didattica in cui utilizzare il computer.

Basandoci sulle nostre conoscenze e partendo soprattutto dalla nostra pratica quotidiana, il modello linguistico da noi scelto è stato quello della semantica generativa del Parisi.

Il tentativo però di "informatizzare" alcuni aspetti del modello parisiense ha fatto nascere in noi grossi dubbi sia di natura ideologica sia di natura pedagogica.

Il rischio di un uso poco corretto del computer è non solo quello di ricadere in un tipo di istruzione meccanizzata, ma anche quello di svilire la professionalità e la funzione del docente.

Tali problemi sono stati motivo di ampio dibattito durante tutta la fase di progettazione del nostro lavoro.

La maggior parte dei "software" didattici attualmente in commercio, infatti, da un lato costringono i ragazzi in una posizione di semplici esecutori e fruitori passivi, con il risultato di un apprendimento di tipo esclusivamente meccanico, dall'altra sviliscono il ruolo del docente, limitando la sua possibilità di intervento.

Proprio per evitare tali rischi siamo giunti alla conclusione che l'insegnante, dopo aver sviluppato con i ragazzi l'intero itinerario didattico, potrà avvalersi delle possibilità offerte dal nuovo mezzo nell'ambito di un testing oggettivo.

Tale uso, se a molti potrà apparire estremamente limitato, offre tuttavia notevoli vantaggi.

L'insegnante facendo eseguire le prove di verifica, sul computer, potrà registrare subito i risultati ottenuti dagli alunni, discutere immediatamente sulle difficoltà da loro incontrate e organizzare in tempi brevi un lavoro di recupero individualizzato.

Questo pedagogicamente rappresenta un grosso vantaggio. Nella nostra pratica didattica, infatti, ci rendiamo conto che spesso dal momento della verifica a quello della discussione sui risultati ottenuti, intercorre un periodo di tempo troppo lungo che spesso vanifica i successivi interventi di recupero.

L'insegnante potrà inoltre, facilmente e con poco dispendio di tempo, ricontrollare i risultati di ogni singola prova e definire così i progressi ottenuti da ogni alunno durante l'intero percorso didattico. Poter, infine, proporre dei quesiti estrapolandoli dall'intero contesto, se da un lato limita la visione d'insieme del problema, dall'altro consente all'alunno di concentrare la sua attenzione sui singoli aspetti della prova. Ciò limitatamente ad un ambito di testing può essere letto in modo positivo, tenendo conto che spesso agli alunni risulta più semplice dare delle risposte ad un problema ben definito e circoscritto.

Riteniamo questi alcuni aspetti da non sottovalutare, ma intendiamo chiarire che quello che presentiamo è solo una proposta e come tale sicuramente discutibile.

È nostra intenzione infatti solo fornire un contributo in un momento in cui il mondo della scuola si sta aprendo ad una nuova rivoluzione culturale.

4.3.1 Ipotesi di lavoro

I giudizi del parlante

“I giudizi del parlante costituiscono la base empirica per la costruzione e la scoperta da parte dei ragazzi della competenza linguistica” (1).

Prima di illustrare il nostro lavoro ci sembra indispensabile, perché sia comprensibile la scelta del modello teorico cui facciamo riferimento, accennare al tema centrale che Parisi tratta nell'elaborazione della sua teoria: i giudizi del parlante.

Intendiamo sottolineare che non affronteremo in modo ampio un aspetto tanto importante della semantica generativa, il nostro scopo infatti, non è quello di analizzare il modello parisiense, ma quello di offrire al lettore indicazioni sulla costruzione di verifiche secondo Unità Didattiche Software e su percorsi didattici, che possono prevedere l'uso di tali verifiche (2).

Fatta questa precisazione cerchiamo di chiarire cosa sono e quale funzione abbiamo i giudizi del parlante.

Se l'obiettivo fondamentale dell'educazione scolastica è quello di “creare le condizioni perché si sviluppi la capacità degli allievi di parlare, di capire, di leggere, di scrivere e di sapere fare tutte queste cose nella varietà più ampia di contesti, di argomenti, di usi” (3), lo scopo di una riflessione sul linguaggio, secondo Parisi, è quello di guidare il ragazzo, partendo dai suoi bisogni e dalle sue capacità, alla “conoscenza scientifica della realtà linguistica” (4), cioè alla possibilità di ricondurre ad una forma di unità i fenomeni linguistici in modo da “poterli comprendere, spiegare e controllare” (5).

La conoscenza linguistica, come ogni conoscenza scientifica, dovrà allora partire da dati empirici e in primo luogo dai “giudizi che colui che parla una lingua sa dare sulle frasi della sua lingua (se sono accettabili e in quel senso, se sono ambigue, se sono parafrasi l'uno dell'altra, e così via)” (6).

Parisi chiama questi i giudizi del parlante ed afferma che: “Vi sono almeno tre proprietà dei giudizi del parlante che ne fanno... i fatti più importanti nello studio del linguaggio. Essi sono i dati empirici (A) più diretti, (B) più dettagliati, (C) più facilmente accessibili per chi voglia studiare la competenza linguistica” (7).

Più diretti in quanto, pur non essendo ancora del tutto definiti i processi attraverso cui il parlante riesce a comprendere le frasi ed a formulare dei giudizi, i meccanismi che si avviano partono dal presupposto che “chi parla una lingua è in grado di esprimersi sulle espressioni della sua lingua”, (8) e quindi forniscono “l'accesso più diretto e privo di ostacoli alla competenza linguistica” (9).

Più dettagliati perché forniscono, rispetto ad altri dati, informazioni più ricche, meno parziali e sommarie sulla competenza linguistica, infatti proprio l'uso “cosciente e

(1) “Per una educazione linguistica razionale” a cura di D. Parisi Il Mulino, p. 415.

(2) Per un approfondimento di tale tematica ricordiamo i testi già citati:

“Per una educazione razionale” a cura di D. Parisi. Il Mulino.

“Elementi di Grammatica” D. Parisi F; Antinucci, Boringhieri.

(3) “Elementi di Grammatica” Op. Cit., p. 250.

(4) Idem, p. 252.

(5) Idem, p. 252.

(6) Idem, p. 249.

(7) “Per una educazione linguistica razionale” Op. Cit., p. 389.

(8) Idem, p. 387.

(9) Idem, p. 392.

sistematico” dei giudizi del parlante ha permesso la descrizione approfondita “di una grande varietà di fenomeni linguistici” (1).

Più facilmente accessibili in quanto chi parla una lingua “è in grado di fornire una massa enorme di giudizi in un tempo relativamente breve, senza apparati o strumenti o condizioni particolari” (2).

Ma l'importanza dei giudizi del parlante può risultare chiara solo quando si analizza il loro fondamentale uso sul piano didattico-pedagogico.

Un modo di utilizzare i giudizi che i ragazzi possono esprimere sulle frasi è quello di far sorgere problemi: “Perché questa frase è accettabile e questa no? Cosa hanno in comune queste due frasi che le rende una parafrasi dell'altra? Perché questa frase è ambigua e quest'altra, pur così simile, ha invece una sola interpretazione? Perché cambiando questa parola la frase diventa inaccettabile, mentre cambiando quest'altra significa una cosa diversa ma resta accettabile?” (3).

Proprio facendo emergere tali problemi l'alunno è portato spontaneamente a “riflettere” sulle sue conoscenze, a scoprire i meccanismi che sottostanno alla vasta gamma dei fenomeni linguistici e ad accedere nel modo più diretto alla “competenza linguistica”.

In tal senso l'avvio alla riflessione sul linguaggio non significa più imposizione di nozioni dall'esterno, mancanza di comprensione, frustrazione, ma scoperta di problemi, ricerca di soluzioni, acquisizione di un metodo scientifico.

Una riflessione sui fenomeni linguistici, così come viene elaborata dal modello parisiense, assume quindi un significato completamente diverso da quello proposto dal modello della “grammatica tradizionale”, non solo per il suo contenuto innovativo, ma anche e soprattutto per il cambiamento didattico pedagogico che è forzatamente richiesto.

La procedura metodologica, che tale studio prevede, serve infatti da un lato ad evitare un apprendimento di tipo meccanico ed un atteggiamento del tutto passivo del ragazzo, dall'altro a modificare il ruolo dell'insegnante nel rapporto educativo.

Per quanto riguarda il primo aspetto bisogna sottolineare che l'alunno viene guidato a scoprire quei meccanismi, che sottostanno ai fenomeni linguistici, attraverso una discussione. Ciò sul piano dell'apprendimento offre notevoli vantaggi. Parisi afferma che “lo sforzo compiuto da ognuno per esercitare una sorta di controllo ragionato sul prodotto delle riflessioni altrui induce non solo ad un alto grado di partecipazione, ma affina le capacità del singolo di analizzare e penetrare l'oggetto in esame, aumentando il controllo sulle proprie riflessioni, ma anche stimolando l'attività di ricerca creativa” (1). Ma come dicevamo questo non è l'unico aspetto innovativo di un diverso approccio all'educazione linguistica, ciò che si ridefinisce infatti è anche il ruolo dell'insegnante. “Nella discussione sui giudizi del parlante, l'insegnante perde lentamente la tradizionale posizione di chi è fuori della mischia perché sa tutto fin dall'inizio... La sua competenza linguistica, in quanto base per esprimere i giudizi, è sostanzialmente uguale a quella dei ragazzi, o almeno, sono molto più le cose “linguistiche” che l'insegnante e i ragazzi sanno insieme, di quelle che sa solo l'insegnante” (2). Anche l'insegnante quindi partecipa in modo attivo alla discussione, ma ciò non significa sminuire o svilire il suo ruolo. Partendo infatti dai dati emersi durante il momento della discussione sarà suo compito quello di guidare gli

(1) “Per una educazione linguistica razionale” Op. Cit., p. 394.

(2) Idem, p. 394.

(3) “Elementi di Grammatica” Op. Cit., p. 256

(4) “Per una educazione linguistica razionale” Op. Cit. P. 404

(5) Idem, p. 407.

alunni all'osservazione ed alla riflessione e di "rendere esplicito e sistematico ciò che il ragazzo sa e sa fare"(1).

Il modello grammaticale viene così progressivamente costruito dagli alunni, cioè appreso, con la costante guida dell'insegnante. Solo in questo modo i ragazzi si apriranno la strada nel modo più diretto alla "competenza linguistica".

Questa descrizione, pur non essendo esaustiva di una tematica così ampia come quella che riguarda i giudizi del parlante, ci è sembrata indispensabile per chiarire la scelta del "modello grammaticale" cui facciamo riferimento nella nostra ipotesi di lavoro. Intendiamo infine sottolineare che il nostro progetto, proprio perché si configura come un'ipotesi, affronta in modo particolare solo l'aspetto relativo al giudizio di accettabilità/non accettabilità delle frasi.

Percorsi didattici sul giudizio di accettabilità/non accettabilità

Nella premessa abbiamo spiegato i motivi che ci hanno spinto a prevedere l'uso del "computer" solo nella fase di verifica, ci rendiamo tuttavia conto che la nostra ipotesi difficilmente risulterebbe leggibile, se non fosse completata da alcune indicazioni sull'intero percorso didattico, che precede il momento conclusivo delle verifiche.

Il lavoro sul giudizio di accettabilità/non accettabilità delle frasi, che rappresenta un primo passo verso la costruzione di quel modello "grammaticale" capace di spiegare in modo organico e coerente i fenomeni linguistici, può essere articolato in tre diverse fasi, ognuna delle quali sviluppa un aspetto particolare di tutta la tematica:

- 1 i giudizi di non accettabilità;
- 2 importanza comunicativa delle regole di accordo;
- 3 importanza semantica dell'ordine delle parole.

Definiremo in modo semplice e schematico, per ciascuna fase di lavoro: l'argomento, gli obiettivi, i compiti dell'insegnante e degli alunni.

I giudizi di non accettabilità

Nella prima fase si affronta il problema della non accettabilità e si definiscono i motivi che rendono una frase non accettabile. Lo scopo di tale fase è quello di ricercare il modo più semplice e più facilmente accessibile per i ragazzi di spiegare i complessi meccanismi, che rendono poco chiaro o che impediscono la comunicazione.

Obiettivo: saper riconoscere quando una frase non è accettabile.

Sottobiettivi:

- saper riconoscere se una frase non è accettabile per:
DISACCORDO GRAMMATICALE;
- saper riconoscere se una frase non è accettabile per:
DISACCORDO DI SIGNIFICATO;
- saper riconoscere se una frase non è accettabile perché:
DISORDINATA;
- saper individuare gruppi di frasi i cui motivi di non accettabilità siano sempre gli stessi.

(1) idem, p. 414.

Compiti dell'insegnante e degli alunni

Per facilitare la lettura abbiamo pensato di rappresentare con uno schema i compiti svolti dall'insegnante e dagli alunni.

Tale procedura sarà utilizzata anche nelle due fasi successive.

Cosa fa l'INSEGNANTE	Cosa fa l'ALUNNO
<ul style="list-style-type: none">— prepara unità didattiche sul problema della non accettabilità e sui motivi che rendono una frase non accettabile— mette a punto materiali e procedure con cui lavorare in classe— discute con gli alunni sui meccanismi che rendono poco chiara o impediscono la comunicazione— aiuta gli alunni a riscontrare la sistematicità e le regolarità che sottostanno a tali meccanismi— fornisce agli alunni un modo convenzionale per definire i motivi che rendono una frase non accettabile, distinguendo tra:<ul style="list-style-type: none">a) disaccordo di significatob) disaccordo grammaticalec) frase disordinata— fornisce agli alunni schede, questionari o esercizi sul problema della "non accettabilità"— chiede agli alunni di esprimere giudizi sulla non accettabilità delle frasi— chiede agli alunni di individuare gruppi di frasi i cui motivi di non accettabilità siano sempre gli stessi	<ul style="list-style-type: none">— discute con l'insegnante e con i compagni sui meccanismi che rendono poco chiara o impediscono la comunicazione— ricerca la sistematicità e le regolarità che sottostanno a tali meccanismi— registra il modo convenzionale per definire i motivi che rendono una frase non accettabile, fornito dall'insegnante, distinguendo tra:<ul style="list-style-type: none">a) disaccordo grammaticaleb) disaccordo di significatoc) frase disordinata— esegue le schede, i questionari o gli esercizi, forniti dall'insegnante— esprime giudizi sulla non accettabilità delle frasi— ricerca gruppi di frasi i cui motivi di non accettabilità siano sempre gli stessi

Importanza comunicativa delle "regole di accordo"

Nella seconda fase si prendono in esame le "regole di accordo".

Lo scopo di tale fase è quello di sottolineare come, variando l'accordo tra predicazione e argomento o tra argomento e argomento di una stessa frase, questa, pur risultando accettabile, assume un significato diverso. Il rispetto delle "regole di accordo" ha evidentemente un'importanza fondamentale ai fini di una corretta comunicazione. Ciò che si vuole sottolineare, in questa fase, non è "l'errore", quanto piuttosto la necessità di costruire una frase rispettando le "regole di accordo", affinché questa possa realmente rispondere allo scopo comunicativo del parlante. Può risultare ora anche più chiaro perché si parla di: importanza comunicativa delle "regole di accordo".

Obiettivo: saper valutare l'importanza comunicativa delle "regole di accordo".

Sottobiettivi:

- saper completare le frasi ed individuare le diverse possibilità di soluzione;
- saper individuare, in base alle diverse soluzioni, gli elementi in accordo;
- saper riconoscere il diverso significato di una frase, se cambiano gli elementi in accordo.

Compiti dell'insegnante e degli alunni

Cosa fa l'INSEGNANTE	Cosa fa l'ALUNNO
<ul style="list-style-type: none">— prepara unità didattiche sul problema delle regole di accordo— mette a punto materiali e procedure con cui lavorare in classe— sottopone agli alunni schede in cui vi siano termini da completare— discute con gli alunni sulle soluzioni da loro proposte— chiede agli alunni di ricercare con quale predicazione o argomento si accorda il termine completato— chiede agli alunni di notare, completando il termine in modo diverso, con quale predicazione o argomento questo si accorda— chiede agli alunni di registrare se ad un diverso completato corrisponde un diverso accordo— aiuta gli alunni a scoprire il significato della frase, a seconda di com'è stata completata— discute con gli alunni sull'importanza comunicativa delle regole di accordo.	<ul style="list-style-type: none">— esegue le schede proposte dall'insegnante, completando i termini— discute con l'insegnante e con i compagni sulle soluzioni trovate— ricerca con quale predicazione o argomento si accorda il termine completato— ricerca, completando il termine in modo diverso, con quale predicazione o argomento questo si accorda— registra se ad un diverso completato corrisponde un diverso accordo— ricerca, con l'insegnante e con i compagni, il significato della frase a secondo di com'è stata completata— discute con l'insegnante sull'importanza comunicativa delle regole di accordo.

Importanza semantica dell'ordine delle parole

Nella terza fase si affronta il problema “dell'ordine delle parole” nella costruzione di una frase. Lo scopo è quello di porre gli alunni nella condizione di scoprire che spesso, variando l'ordine delle parole all'interno di una frase, questa assume un diverso significato. Ma poiché ogni frase esprime un preciso messaggio, che sottende le intenzioni comunicative del parlante, queste devono essere costruite in modo da riflettere pienamente lo scopo comunicativo.

Lavorando direttamente sulle frasi variando, ad esempio, la posizione di un elemento (o gruppi di elementi) i ragazzi comprenderanno meglio cosa si intende per: “importanza semantica dell'ordine delle parole”.

Obiettivo: saper valutare l'importanza semantica dell'ordine delle parole.

Sottobiettivi:

- saper individuare quali frasi restano accettabili, quando un elemento (o gruppo di elementi) della frase cambia posto;
- saper riconoscere se la frase assume un diverso significato, quando un elemento (o gruppi di elementi) cambia di posto.

Compiti dell'insegnante e degli alunni

Cosa fa l'INSEGNANTE	Cosa fa l'ALUNNO
<ul style="list-style-type: none"> — prepara unità didattiche sul problema dell'importanza semantica dell'ordine delle parole — mette a punto materiali e procedure con cui lavorare in classe — fornisce agli alunni schede o esercizi — chiede agli alunni di spostare un elemento (o gruppi di elementi) in tutti i punti possibili della frase — discute con gli alunni sulle frasi da loro formulate invitandoli ad esprimere su ciascuna il giudizio di "accettabilità/non accettabilità" — chiede agli alunni di elencare tutte le frasi accettabili — chiede agli alunni di confrontare le frasi da loro formulate e giudicate accettabili con la frase data e di verificare se il significato è rimasto lo stesso o è cambiato — discute con gli alunni sul significato delle frasi — discute con gli alunni sull'importanza semantica dell'ordine delle parole 	<ul style="list-style-type: none"> — sposta un elemento (o gruppo di elementi) in tutti i punti possibili della frase — discute con l'insegnante e con i compagni sulle frasi da lui formulate ed esprime su ciascuna il giudizio di "accettabilità/non accettabilità" — elenca tutte le frasi accettabili come richiesto dall'insegnante — confronta le frasi formulate e giudicate accettabili con la frase data e verifica se il significato è rimasto lo stesso o è cambiato — discute con l'insegnante e con i compagni sul significato delle frasi — discute con l'insegnante e con i compagni sull'importanza semantica dell'ordine delle parole

L'insegnante concluderà ciascuna fase di lavoro con una prova di verifica che chiameremo formativa e l'intero itinerario didattico con una verifica sommativa. Le verifiche formative risulteranno fondamentali per: verificare gli obiettivi raggiunti da ogni singolo alunno, registrare le eventuali difficoltà incontrate, strutturare lavori di recupero individualizzati. La verifica sommativa offrirà all'insegnante dei dati sui risultati ottenuti dagli alunni e avrà solo una funzione riassuntiva rispetto al livello di apprendimento da loro raggiunto.

Le verifiche:

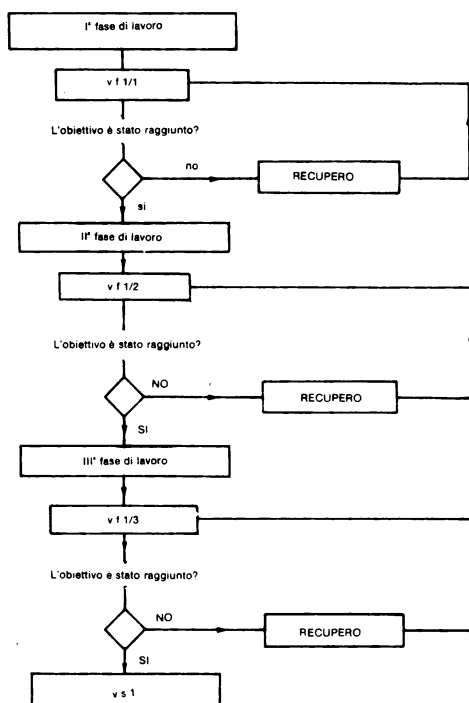
vf 1/1

vf 1/2

vf 1/3

vs 1

sono state da noi realizzate con programma BASIC. Nelle pagine seguenti saranno spiegate le procedure adottate per strutturare tali verifiche e saranno fornite tutte le indicazioni per svolgere le prove sul computer. Qui presenteremo invece il diagramma di flusso che sintetizza il percorso didattico che l'insegnante dovrà seguire:



4.3.2 Struttura generale delle verifiche

Spesso ai non addetti ai lavori la lettura delle pagine dedicate all'illustrazione dei programmi risulta noiosa e alquanto ostica.

L'uso di un linguaggio "artificiale", sia esso basic o altro, crea infatti non pochi problemi di comprensione a chi è privo di tale "codice comunicativo". La stessa diffidenza di molti insegnanti, nel prevedere l'utilizzo del computer nella pratica didattica, è spesso dovuta alla "mancanza" di adeguate conoscenze.

Partendo dalla convinzione che, al momento attuale, si possa e si debba richiedere ai docenti una diversa professionalità, abbiamo cercato di rendere, nei limiti delle nostre capacità, più chiara e comprensibile la visione dell'intero progetto e la metodologia da noi adottata nel realizzare i software didattici.

Il metodo seguito nella elaborazione dei programmi presenta molte analogie con il tipo di programmazione a cui siamo abituati, la stessa rappresentazione grafica del MENU PRINCIPALE ne è la conferma.

Non parleremo quindi nè di "programmazione strutturata" nè di "pseudo-codice" e ciò se da un lato meravighierà gli esperti del settore, dall'altro renderà più semplice la comprensione del nostro lavoro.

Tutte le verifiche saranno costituite dai seguenti sottoprogrammi:

- 1 GOSUB
- 2 MENU
- 3 INIZIALIZZARE
- 4 INSERIRE

- 5 LISTARE
- 6 CORREGGERE
- 7 STAMPARE
- 8 REGISTRARE
- 9 ESEGUIRE

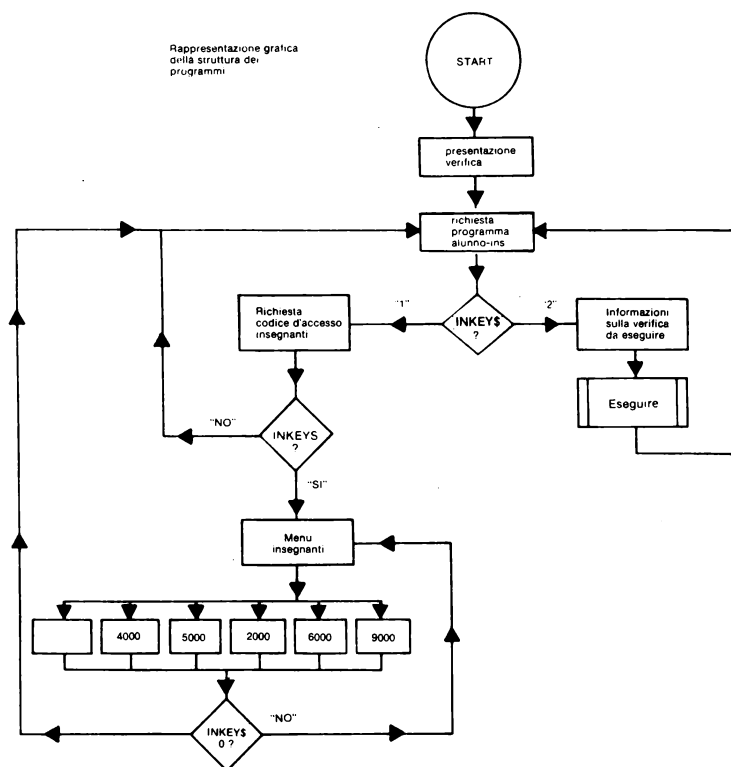
Questi presentano tutti la stessa “impaginazione video” e le stesse modalità operative, mentre si diversificano per il tipo di argomentazione trattata. Pertanto la spiegazione risulterà particolareggiata nella prima verifica vf1/1 e più schematica nelle altre, fatta eccezione per i programmi INIZIALIZZARE-INSERIRE-ESEGUIRE, che presentando variazioni, saranno illustrati di volta in volta.

Quindi laddove si troverà solo il programma in BASIC è implicito che, per il commento, si farà riferimento al paragrafo corrispondente della vf 1/1.

Le verifiche da noi proposte sono già complete, l'insegnante potrà controllarne il contenuto, ovvero le frasi, con il sottoprogramma LISTARE, e decidere se utilizzarle così come sono state strutturate.

Bisogna ricordare che per accedere ai sottoprogrammi è indispensabile comporre il “CODICE D'ACCESSO” al MENU PRINCIPALE, tale codice sarà conosciuto solo dall'insegnante. Questo accorgimento, si è reso necessario per evitare la cancellazione, magari involontaria, da parte di altri della prova di verifica preparata dal docente.

L'impostazione adottata per tutte le prove di verifica è rappresentata nel seguente schema:



4.3.3 vf 1/1 - Giudizi di non accettabilità

I GOSUB

Abbiamo preferito trattare i GOSUB isolatamente e all'inizio di ogni verifica per consentire una lettura più spedita e lineare dei programmi ed evitare continue interruzioni, da parte del lettore, per ricercare le funzioni da questi svolte.

Cosa significa il termine GOSUB e quali sono i vantaggi offerti da queste sottoprocedure?

La parola-chiave GOSUB, in BASIC, sta per "GO TO SUBROUTINE" ed, una traduzione letterale, potrebbe essere "VAI AD ESEGUIRE LA SOTTOPROCEDURA E RITORNA AL PROGRAMMA PRINCIPALE". La richiesta viene effettuata dalla parola-chiave GOSUB e il ritorno al programma è dato dalla parola-chiave RETURN. Questi due comandi non possono essere usati separatamente; ciò infatti comporterebbe una segnalazione di errore da parte del computer ed una interruzione del programma.

Non vi è limite al numero di istruzioni che una sottoprocedura può contenere. Necessario sarà, però, concludere le istruzioni con il comando RETURN all'ultima linea.

Il ricorso ad un sottoprogramma offre grossi vantaggi:

- disporre, all'interno di un programma, di istruzioni di salto che permettono l'esecuzione di un sottoprogramma;
- poterlo richiamare ogni volta che lo si consideri utile e necessario;
- evitare la ripetizione di istruzioni che occuperebbero parte della memoria disponibile.

Alcuni GOSUB risolvono aspetti colloquiali, contengono cioè informazioni o comunicazioni video per l'utente.

Questi GOSUB sono comuni a quasi tutte le verifiche e saranno spesso richiamati durante le fasi di costruzione ed esecuzione delle prove.

GOSUB 9500

Programma in BASIC (9500÷9504)

```
9500>REM GO SUB
9501 IF INKEY$(">") THEN GO TO 95
9502 IF INKEY$="" THEN GO TO 950
9503 LET X$=INKEY$: IF CODE (X$)
<>115 AND CODE (X$) <>110 THEN BE
EP 1,10: GO TO 9500
9504 RETURN
```

Queste istruzioni consentono di:

- rallentare la velocità di lettura del computer quando viene premuto un tasto qualsiasi controllato dalla funzione INKEY\$ (9501÷9502);
- controllare la scelta operata sul "s" e sul "n" e di memorizzarla in X\$ (9503).

Operata la scelta e ricevuto il dato, quest'ultimo viene riconosciuto e memorizzato: se risulterà diverso dal codice di memoria del SI o del NO (s=115; n=110) emetterà un segnale sonoro, richiamando l'attenzione dell'utente sui tasti da utilizzare; viceversa, riconosciuto uno dei due, ritornerà all'esecuzione del programma principale ovvero alla linea d'istruzione successiva a quella del comando GOSUB.

Questo GOSUB, comune a tutte le unità software e presente in molte procedure contenute nei programmi è risultato indispensabile per evitare errori sulla scelta da MENU.

Il programma INIZIALIZZARE, ad esempio, cancella, come vedremo in seguito, i dati della verifica, cioè le frasi già esistenti e consente l'impostazione di una nuova prova. Se ciò non fosse nelle intenzioni dell'insegnante, nel caso quindi di una scelta errata del programma, è necessario poter correggere l'errore e consentire il ritorno al MENU PRINCIPALE

GOSUB 9505

```
9505>REM GO SUB
9506 IF INKEY$<>"" THEN GO TO 95
9507 IF INKEY$="" THEN GO TO 950
9508 LET X$=INKEY$
9509 RETURN
```

Programma in BASIC (9505÷9509)

Questo GOSUB rallenta l'esecuzione del programma e ha lo scopo di memorizzare in X\$ il dato richiesto dalla fase operativa del programma principale. Nelle nostre Unità Software è stato usato spesso anche per programmi diversi e tutte le volte in cui il dato richiesto doveva essere di un solo carattere (per numeri di una cifra, per una lettera dell'alfabeto).

La funzione INKEY\$ obbliga il computer a leggere un carattere per volta; se nessun tasto viene premuto (""), il computer è in attesa del dato come se si trovasse di fronte ad un'istruzione di INPUT.

Premuto il tasto, il dato viene memorizzato con l'istruzione LETx\$=INKEY\$.

GOSUB 9550/GOSUB 9600

Programma in BASIC (9550÷9552/9600÷9602)

```
9550>REM GO SUB
9551 PRINT FLASH 1;AT 21,0;S$
9552 PAUSE 0:CLS : GO TO 180
9600 REM GO SUB
9601 PRINT FLASH 1;AT 21,0;S$
9602 PAUSE 0:CLS : RETURN
```

Questi GOSUB servono per conservare una comunicazione video, precedentemente memorizzata in S\$.

Richiamando il sottoprogramma, comparirà sul video lampeggiando, alla riga 21 del quadro, il contenuto di S\$ (PREMI UN TASTO PER CONTINUARE). La differenza e la necessità di averli entrambi sta nel fatto che il GOSUB 9550 consente il ritorno al MENU con il comando GOTO 180; il secondo, dopo la comunicazione video, consentirà di riprendere l'esecuzione del sottoprogramma.

Questo è un caso in cui il GOSUB viene costruito per risolvere un aspetto colloquiale del programma.

GOSUB 9650

Ogni volta che verrà espresso un giudizio di NON ACCETTABILITÀ, sia durante la fase di inserimento delle frasi da parte dell'insegnante, sia durante l'esecuzione della prova da parte dell'alunno, sul video apparirà la seguente comunicazione:

```

          PERCHE'
    LA FRASE NON E' ACCETTABILE ?

SE DISORDINATA ----->PREMI 1
SE PER DISACCORDO
GRAMMATICALE ----->PREMI 2
SE PER DISACCORDO
DI SIGNIFICATO ----->PREMI 3

```

È evidente il vantaggio offerto, in questo caso, dal GOSUB: la verifica infatti è costituita da più frasi che risultano non accettabili, per ognuna di queste unità si sarebbero dovute ripetere le istruzioni di stampa della comunicazione video.

Programma in BASIC (9650÷9652)

```

9650>REM GO SUB
9651 PRINT AT 10,12,"PERCHE' TA
B 2:0$;"?": PRINT : PRINT : PRIN
T : TAB 0;"SE DISORDINATA -----
->PREMI 1": PRINT : PRINT : TAB 0
;"SE PER DISACCORDO": TAB 0;"GRAM
MATICALE ----->PREMI 2": PR
INT : PRINT : TAB 0;"SE PER DISAC
CORDO": TAB 0;"DI SIGNIFICATO ---
----->PREMI 3"
9652 RETURN

```

L'istruzione di stampa (PRINT) e l'uso della parola-chiave TAB consentono di allineare, andando a capo, le righe di scrittura del messaggio.

GOSUB 9700

Il numero delle frasi e la lunghezza massima di ogni frase, espressa in numero di caratteri disponibili, saranno visualizzati dalla seguente comunicazione video. Per esempio:

```

QUESTA VERIFICA E' COSTITUITA
DA 20 FRASI.

OGNI FRASE DISPONE AL MAX. DI
92 CARATTERI.

```

Questo tipo di informazioni è particolarmente utile all'insegnante che, in fase di costruzione della prova, potrà meglio scegliere le frasi da inserire.

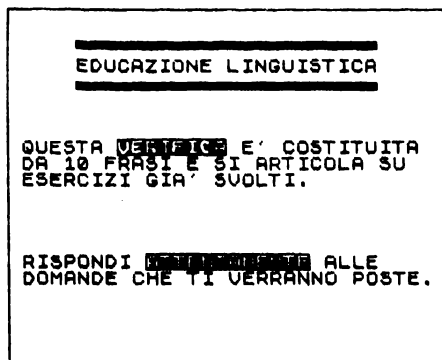
Programma in BASIC (9700÷9702)

```
9700>REM GO SUB
9701: PRINT AT 9,0;" QUESTA VERI
FICA E' COSTITUITA DA ",X," FR
ASI." PRINT : PRINT TAB 0;" OGN
I FRASE DISPONE AL MAX. DI 92
CARATTERI."
9702 RETURN
```

C'è da notare che il numero delle frasi è dato dalla variabile x questa potrà ogni volta mutare, sia impostando una nuova verifica (non più di 10, ma, ad esempio, di 20 frasi), sia ampliando quella già esistente (inserendo cioè di volta in volta nuove frasi).

GOSUB 9710

L'alunno sarà informato sulla natura della verifica, che dovrà sostenere, con la seguente comunicazione video. Per esempio:



Programma in BASIC (9710÷9730)

```
9710>REM GO SUB
9712 PRINT AT 2,5;" PAPER 6;"
S; PAPER 6;"EDUCAZIONE LINGUISTI
CA"
9715 PRINT AT 4,5;" PAPER 6;"
9720 PRINT AT 8,1;"QUESTA VERIFI
CA E' COSTITUITA" TAB 1;"DA ";S;
" FRASI E SI ARTICOLA SU " TAB 1
;"ESERCIZI GIA' SVOLTI."
9725 PRINT AT 15,1;"RISPONDI ATT
ENTAMENTE ALLE " TAB 1;"DOMANDE
CHE TI VERRANNO POSTE."
9730 RETURN
```

Le istruzioni in BASIC del programma per la loro semplicità non necessitano di un particolare commento.

GOSUB 9750

Questo GOSUB consente di visualizzare alla riga 20 del video, lampeggiandola, la comunicazione: "PREMI UN TASTO PER CONTINUARE"

Programma in BASIC (9750÷9752)

```
9750>REM GO SUB  
9751 PRINT FLASH 1:AT 20.0:"INSE  
9752 IL DATO." PREMI ENTER."  
9752 RETURN
```

La parola-chiave FLASH permettere di lampeggiare il messaggio.

Tale accorgimento è stato usato per richiamare l'attenzione, di chi opera sulle varie sequenze.

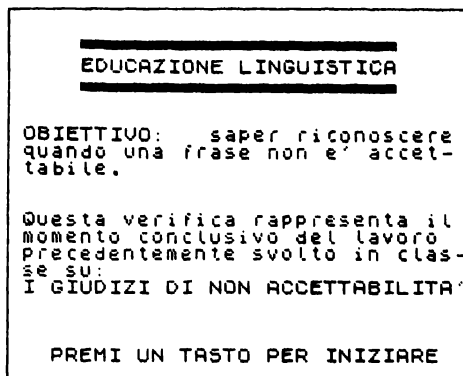
MENU

Il programma MENU si compone di tre parti:

- Presentazione verifica;
- codice d'accesso;
- menu insegnanti.

Presentazione verifica

La verifica inizia con una comunicazione video che si presenta nel seguente modo:



La possibilità di utilizzo di un qualsiasi programma su computer ed, in particolare, di un software didattico risulterà maggiore, quanto più chiare saranno le informazioni fornite all'utente.

L'aspetto "colloquiale" dovrà quindi essere particolarmente curato ai fini di un buon uso dell'intero lavoro.

È per tale motivo che nelle nostre verifiche vi sono spesso comunicazioni video all'utente.

Programma in BASIC (1÷50)

```
1>REM MENU
5:CLS : BORDER 1: PAPER 7: IN
K 0: CLS
10 PRINT AT 2,5: PAPER 6: "
5: PAPER 6: "EDUCAZIONE LINGUISTI
CA"
20 PRINT AT 4,5: PAPER 6: "
30 PRINT AT 7,1: "OBIETTIVO:
saper riconoscere quando una fr
ase non e' accet- tabile."
40 PRINT AT 12,1: "Questa verif
ica rappresenta il momento conc
lusivo del lavoro precedenteme
nte svolto in clas- se su: TAB
1: "I GIUDIZI DI NON ACCETTABILIT
A"
50 PRINT AT 20,3: "PREMI UN TAS
TO PER INIZIARE": PAUSE 0: GO TO
60
```

Le linee d'istruzione relative a questa prima parte del programma MENU non presentano particolari difficoltà. Si tratta di ordini di stampa (PRINT) e della collocazione del messaggio sul video (AT); i numeri che seguono "AT" indicano la riga e la colonna da cui deve iniziare la stampa.

Le istruzioni alla linea 5 creano un riquadro sullo schermo, di colore azzurro (BORDER 1) all'interno del quale, su fondo bianco (PAPER 7), è visualizzata la scritta con il colore nero (INK 0).

L'istruzione inizia con la parola-chiave CLS che serve a cancellare dallo schermo tutto ciò che era precedentemente scritto; ogni comando è separato dai due punti (:).

L'ultima linea di programma presenta due comandi PAUSE e GOTO.

Il comando PAUSE, seguito da un numero, determina il tempo di permanenza della scritta sullo schermo (nel nostro caso 0 corrisponde ad un tempo infinito e solo premendo un tasto qualsiasi si passa al comando successivo).

La parola-chiave GOTO, spesso usata, fa "saltare" il programma alla linea dell'istruzione voluta. In seguito vedremo l'importanza di questo comando operativo.

Codice d'accesso

PER INIZIARE LA VERIFICA FARSI RICONOSCERE
PER GLI INSEGNANTI----- 1
PER GLI ALUNNI----- 2

Con questa informazione parte il programma relativo al codice d'accesso. Abbiamo sottolineato l'importanza di questo accorgimento, ma è bene ricordare che

questo è risultato necessario soprattutto per impedire l'accesso ai ragazzi di quelle funzioni, presenti nel MENU PRINCIPALE.

Dopo aver premuto il tasto 1 viene richiesto all'insegnante il codice d'accesso



SCRIVI IL CODICE D'ACCESSO

Il codice di accesso in questo programma è “.” PUNTO.

Se viene premuto un tasto diverso da “.” (PUNTO) compare la scritta:



SCRIVI IL CODICE D'ACCESSO

CODICE SCONOSCIUTO

Per modificare il codice di accesso devono essere modificate le linee 165 e 170.

Programma in BASIC (55÷170)

Il programma che consente la realizzazione di questa seconda parte è il seguente:

```
55:PRINT AT 20,7: PAPER 6:"COD
ICE SCONOSCIUTO": PAUSE 150
60:PAUSE 100:CLS: PLOT 5,75:
DRAW PAPER 6,+245,0: DRAW PAPER
6,0,-40: DRAW PAPER 6,-245,0: C
RAW PAPER 6,0,+40
70:PLOT 6,99: DRAW PAPER 6,+24
5,0: DRAW PAPER 6,0,+40: DRAW PA
PER 6,-245,0: DRAW PAPER 6,0,-40
75:PRINT AT 7,2:"PER GLI INSEG
NANTI-----": PAPER 4:"1"
80:PRINT AT 15,2:"PER GLI ALUN
NI-----": PAPER 2: INK 7
:"2"
```

```

85 PRINT AT 1,4:"PER INIZIARE
LA VERIFICA": PRINT TAB 7;"FARSI
RICONOSCERE"
90 GO SUB 9505
100 IF CODE X$=49 THEN GO TO 15
0
110 IF CODE X$=50 THEN GO TO 80
00
120 IF CODE X$<>49 AND CODE X$<
50 THEN BEEP 1,10: GO TO 90
150 CLS: PRINT AT 10,3: FLASH
1;"SCRIVI IL CODICE D'ACCESSO"
160 GO SUB 9505
165 IF CODE X$=46 THEN GO TO 18
0
170 IF CODE X$<>46 THEN BEEP 1,
10: GO TO 55

```

Le linee d'istruzione che vanno da 60 a 85 contengono comandi di grafica e le due scritte con i numeri necessari al riconoscimento dell'utente. Le parole-chiave PLOT e DRAW disegnano delle linee sul video, che racchiudono in riquadri separati le due indicazioni.

La linea 90 rimanda al GOSUB 9505; esaminiamo quindi le istruzioni successive.

Tutti i caratteri presenti sulla tastiera di un computer, da quelli numerici a quelli alfanumerici, dai segni grafici agli stessi comandi sono conservati nella memoria ROM dell'elaboratore in codice numerico. La funzione CODE serve a ricavare il valore numerico del carattere.

Le linee 100 e 110 verificano se il tasto premuto, memorizzato in X\$, corrisponde a 1 (numero di codice 49) o a 2 (numero di codice 50). Nel primo caso il programma passa alla linea 150, visualizzando la richiesta del codice d'accesso; nel secondo passa alle istruzioni che partono dalla linea 8000 (inizio del programma ESEGUIRE). La linea d'istruzione 120 con un segnale sonoro (BEEP) avvisa che sono stati digitati tasti diversi da "1" o "2" (cioè da 49 o 50). Le ultime due linee (165 e 170) verificano se il codice d'accesso è corretto. Se questo risulterà diverso, il programma ritornerà alla linea 55 che informa dell'inesattezza del codice.

Menu insegnanti (menu principale)

Sul video comparirà il seguente MENU:

EDUCAZIONE <u>LINGUISTICA</u>	PER CONTROLLARE LA VERIFICA
VERIFICA	LISTARE-----5
GIUDIZI NON ACCETTABILITÀ	CORREGGERE----2
	STAMPARE-----6
PER COSTRUIRE LA VERIFICA	PER REGISTRARE LA VERIFICA---9
INIZIALIZ ZARE-----4	
INSERIRE FRASI-----1	COMPLETATE LE OPERAZIONI PREMI----->0

Come si può osservare, le funzioni presenti nel MENU sono state inserite in quattro spazi ben definiti, ciò per consentire una lettura ed un uso più immediato dei programmi cui l'insegnante può accedere:

Il nostro intento è stato quello di evidenziare, così come avviene nella nostra pratica quotidiana, le diverse fasi di lavoro:

- per costruire la verifica (inizializzare-inserire);
- per controllare la verifica (listare-correggere-stampare);
- per registrare la verifica.

Programma in BASIC (180÷510)

```

180>CLS PRINT AT 1,1: PAPER 6
:EDUCAZIONE PRINT TAB 1, PAPER 6
:ER 6; "LINGUISTICA"
190 PRINT AT 3,1: "-----"
200 PLOT 5,5
210 DRAW 0,+100: DRAW +100,0: D
RAW 0,-100: DRAW -100,0
220 PLOT 115,50
230 DRAW +135,0: DRAW 0,+40: DR
AW -135,0: DRAW 0,-40
240 PLOT 115,95
250 DRAW +135,0: DRAW 0,+75: DR
AW -135,0: DRAW 0,-75
320 PRINT AT 4,1: "VERIFICA" PR
INT : PRINT TAB 1: PAPER 6: "GIUD
IZI NON " PRINT TAB 1, PAPER 6
"ACCETTABILITA" PLOT 108,121:
PLOT 109,120
330 PRINT AT 10,1: "PER COSTRUIR
E" PRINT AT 11,1: "LA VERIFICA"
340 PRINT AT 14,1: "INIZIALIZZ" A
T 15,1: "ZARE-----" PAPER 2: I
NK 7: "4" PRINT AT 17,1: "INSERIR
E" PRINT AT 18,1: "FRASI-----"
PAPER 2: INK 7: "1"
350 PRINT AT 12,15: "PER " "REGI
STRARE" PRINT AT 14,15: "LA VERI
FICA-----" PAPER 2: INK 7: "3"
360 PRINT AT 1,15: "PER " "CONTR
OLLARE" PRINT AT 2,15: "LA VERIF
ICA"
380 PRINT AT 4,15: "LISTARE-----
-----" PAPER 2: INK 7: "5" PRINT A
T 6,15: "CORREGGERE-----" PAPER 2
: INK 7: "2" PRINT AT 8,15: "STAM
PARE-----" PAPER 2: INK 7: "6"
450 PRINT AT 13,16: PAPER 6: "CO
MPLETATE LE" PRINT AT 19,16: PA
PER 6: "OPERAZIONI"
460 PRINT AT 20,16: PAPER 6: "PR
EMI-----"
470 GO SUB 9505
480 IF CODE *$=57 OR CODE *$=48
THEN BEEP 1,10 GO TO 470
485 IF CODE *$=51 OR CODE *$=55
OR CODE *$=56 THEN BEEP 1,10 G
O TO 470
487 IF CODE *$=48 THEN GO TO 60
490 LET A=VAL *$
500 GO TO 1000+A
510 GO TO 1

```

Le considerazioni da fare riguardano solo le ultime linee di programma, essendo le prime, da 180 a 460, di grafica e di stampa.

La scelta delle funzioni presenti nel MENU è data dai numeri indicati dalle frecce.

Per evitare che siano battuti tasti diversi da quelli previsti sono state inserite istruzioni di controllo (linee 480,485, 487). Queste verificano, attraverso i numeri di codice, che il tasto premuto sia quello giusto. La linea 480 esclude tutti i tasti non compresi tra 0 (numero di codice 48) e 9 (numero di codice 57); la linea 485 non consente la scelta dei tasti 3 (numero di codice 51), 7 (numero di codice 55) e 8 (numero di codice 56) poiché non previsti nel MENU.

In entrambe le istruzioni, l'eventuale errore di battitura è sottolineato da un segnale sonoro (BEEP 1,10).

La linea 487 permette di ritornare alla richiesta di identificazione, una volta ultimate le operazioni decise dall'insegnante.

Le osservazioni da fare su questa parte del MENU sono relative alle linee 490 e 500. Poiché il tasto premuto viene memorizzato in una variabile di tipo stringa, si è reso necessario l'uso di una nuova funzione, VAL, che trasforma il carattere stringa in valore numerico e lo conserva in una memoria q. Quest'ultima, operando la relazione $1000 * q$ andrà alla ricerca del sottoprogramma da eseguire.

Per costruire la verifica

L'insegnante che voglia costruire una nuova verifica dovrà utilizzare i seguenti sotto programmi, in ordine di successione:

- INIZIALIZZARE;
- INSERIRE FRASI.

Inizializzare

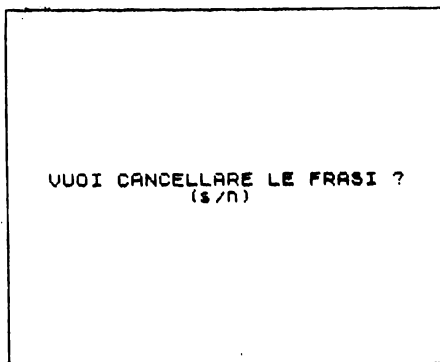
Questo sottoprogramma è di fondamentale importanza poiché consente di cancellare le frasi della prova di verifica già strutturate e di predisporre la macchina a ricevere i nuovi dati. Esso risulta suddiviso in due parti:

- a) dimensionamento della verifica;
- b) individuazione e dimensionamento delle variabili.

a. Dimensionamento della verifica

La pratica ci ha dimostrato che non sempre la stessa prova di verifica può adattarsi ai livelli di partenza della classe o dei singoli allievi, pertanto abbiamo ritenuto utile non predeterminare il numero delle frasi, costituenti la prova, e lasciare all'insegnante la possibilità di ampliare o ridurre la verifica e di utilizzare o meno le frasi da noi inserite.

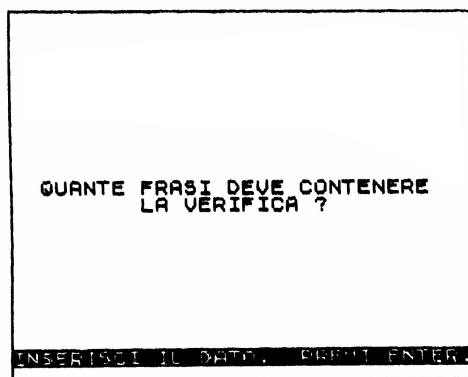
Scegliendo il sottoprogramma INIZIALIZZARE comparirà sul video la seguente scritta:



La richiesta di riconferma del programma scelto è necessaria per consentire all'insegnante di ritornare al MENU nel caso avesse erroneamente premuto il tasto 4.

Solo rispondendo "s", il programma continua la sua funzione, in caso contrario tornerà al MENU PRINCIPALE, consentendo di effettuare una nuova scelta.

Successivamente comparirà la seguente comunicazione:

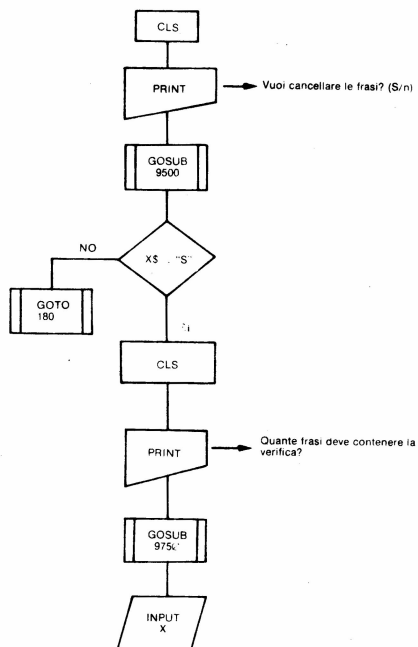


L'insegnante a questo punto deve inserire il dato, che definisce il numero delle frasi, costituenti la prova di verifica.

Ricevuto il nuovo dato il computer cancella il contenuto presente nella memoria RAM, predisponendola al nuovo dimensionamento.

Diagramma di flusso

Diagramma di flusso



Programma in Basic (4000÷4080)

```
4000>REM INIZIALIZZARE
4010 CLS : PRINT AT 10,3:"VUOI C
ANCELLARE LE FRASI ?";TAB 13;"(S
/N)
4020 GO SUB 9500
4040 IF X$="S" THEN GO TO 4060
4050 GO TO 150
4060 CLS
4070 PRINT AT 10,3:"QUANTE FRASI
DEVE CONTENERE";TAB 9;"LA VERIF
ICA ?"; GO SUB 9750
4080 INPUT X
```

Come si può osservare troviamo una nuova parola-chiave INPUT alla linea 4080, seguita da un'incognita senza il simbolo "\$" (stringa). Si tratta in questo caso di una variabile che può contenere solo caratteri numerici.

Il comando INPUT, usato spessissimo, lascia il computer in attesa del dato e non consente la prosecuzione del programma.

b. Individuazione e dimensionamento delle variabili

La seconda parte del sottoprogramma INIZIALIZZARE dimensiona le memorie che dovranno contenere i nuovi dati.

L'individuazione e la definizione delle variabili è subordinata al numero dei dati che si vogliono registrare.

Pertanto per ciascuna informazione che si intende ottenere, o fornire all'intero sistema, corrispondono altrettanti "indirizzi di memoria".

Poiché la verifica su I GIUDIZI DI NON ACCETTABILITÀ si basa sull'individuazione delle caratteristiche comunicative delle frasi e sui motivi che le rendono non accettabili sarà opportuno definire gli elementi che dovranno essere inseriti:

- conoscere il giudizio di accettabilità/non accettabilità espresso sia dall'insegnante sia dall'alunno;
- conoscere il giudizio sui motivi di non accettabilità espresso sia dall'insegnante sia dall'alunno;
- conoscere gli eventuali errori commessi dall'alunno durante la verifica.

Legenda delle variabili

VARIABILE	FUNZIONE	DIMENSIONE DELLA MEMORIA
a\$(x,92)	conserverà la frase inserita dall'insegnante	x:=n° max. di frasi della verifica 92:=n° max. di caratteri disponibili per frase
b\$(x,1)	conserverà la risposta dell'insegnante relativa al giudizio di accettabilità sulla frase (s/n)	x:=n° max. di frasi della verifica 1:= n° carattere disponibile (s o n)
c\$(x,1)	conserverà la risposta dell'insegnante relativa al motivo di non accettabilità sulla frase	x:=n° max. di frasi della verifica 1:=n° carattere disponibile

/segue/

VARIABILE	FUNZIONE	DIMENSIONE DELLA MEMORIA
z\$(x,92)	conserverà la frase su cui è stato espresso un erroneo giudizio da parte dell'alunno.	vedi a \$
y\$(x,1)	conserverà la risposta dell'alunno relativa al motivo di non accettabilità sulla frase.	vedi c\$
i\$(35)	conserverà i dati relativi alunno (cognome, nome, classe)	35:= n° max. di caratteri disponibili (1 TO 24)=n° max di caratteri disponibili per conservare cognome e nome dell'alunno (26 TO 35):=n° max di caratteri disponibili per conservare la classe dell'alunno.

Programma in BASIC (4200÷4900)

```

4200>DIM a$(x,92)
4201 DIM b$(x,1)
4202 DIM c$(x,1)
4203 DIM z$(x,92)
4204 DIM y$(x,1)
4215 DIM i$(35)
4300 LET m$="LA FRASE E' ACCETTA
BILE "
4301 LET n$="LA FRASE NON E' ACC
ETTABILE "
4302 LET p$="PERCHE' DISORDINATA
"
4303 LET q$="PER DISACCORDO GRAM
MATICALE "
4304 LET r$="PER DISACCORDO DI S
IGNIFICATO "
4305 LET s$=" PREMI UN TASTO PE
R CONTINUARE "
4805 CLS
4890 GO SUB 9700
4900 GO TO 9550

```

La parola-chiave che predispone gli indirizzi di memoria è DIM seguita da una variabile.

Per le altre istruzioni presenti nel programma non sono necessarie ulteriori spiegazioni, in quanto già precedentemente illustrate.

L'unica osservazione da fare è che alcuni aspetti colloquiali sono stati memorizzati in variabili-stringa, per maggiore comodità.

Inserire frasi

Decisa l'impostazione della verifica e definiti gli elementi che caratterizzeranno la prova (4200÷4900), l'insegnante potrà inserire le frasi, dopo averne determinato il numero.

La verifica può essere costituita da un numero x di frasi, ciò consentirà all'insegnante di strutturare una prova adeguata ai livelli di ogni singolo alunno e sulla base delle difficoltà emerse durante il lavoro.

Il programma si sviluppa in due fasi:

- a) Determinazione del numero delle frasi da inserire;
 - b) inserimento frasi.
- a) Determinazione delle frasi da inserire

Premendo il tasto corrispondente al numero 1, comparirà sul video la seguente scritta:

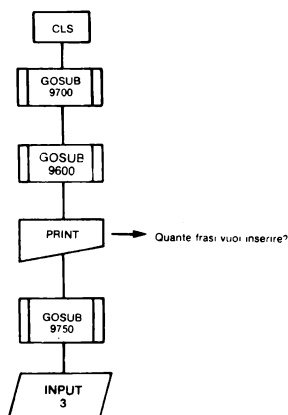
```

QUESTA VERIFICA E' COSTITUITA
DA 20 FRASI.

OGNI FRASE DISPONE AL MAX. DI
92 CARATTERI.

PREMI UN TASTO PER CONTINUARE
  
```

Diagramma di flusso



Programma Basic (1000÷1101).

```

1000>REM INSERIRE
1010 CLS
1020 GO SUB 9700
1030 GO SUB 9600
1040 PRINT AT 10,2;"QUANTE FRASI
VUOI INSERIRE ?": GO SUB 9750
1101 INPUT "
  
```

Queste istruzioni non presentano nuove parole-chiave e rimandano a GOSUB già illustrati; non necessitano quindi di particolari commenti.

b) Inserimento frasi

Durante questa fase l'insegnante dovrà inserire con la frase anche i dati di ciascuna, relativi a:

- la frase è accettabile;
- la frase non è accettabile;
- la frase non è accettabile perché disordinata;
- la frase non è accettabile per disaccordo grammaticale;
- la frase non è accettabile per disaccordo di significato.

La presenza di tanti dati da inserire ci ha posto subito alcuni problemi:

- inserire prima l'elenco delle frasi e poi i dati relativi a ciascuna o operare contemporaneamente l'inserimento?
- come controllare l'esattezza dei dati inseriti?
- come controllare una corretta divisione delle parole nell'andare a capo?

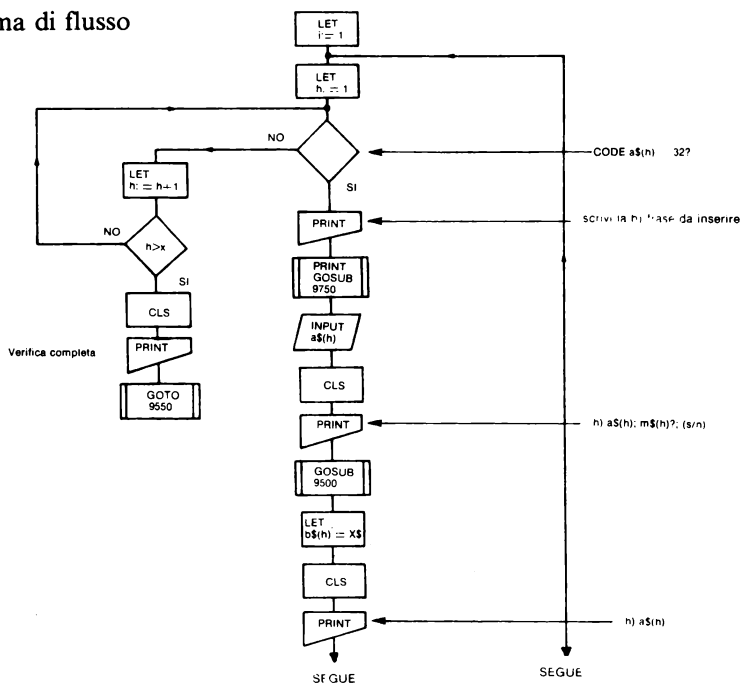
L'ultimo punto è stato risolto indicando con una freccia, rivolta verso l'alto (1) e visualizzata in basso a destra, il margine della riga su cui è possibile scrivere. In questo modo, l'insegnante, sapendo che in corrispondenza del simbolo la parola verrà divisa, potrà meglio controllare l'aspetto ortografico della frase.

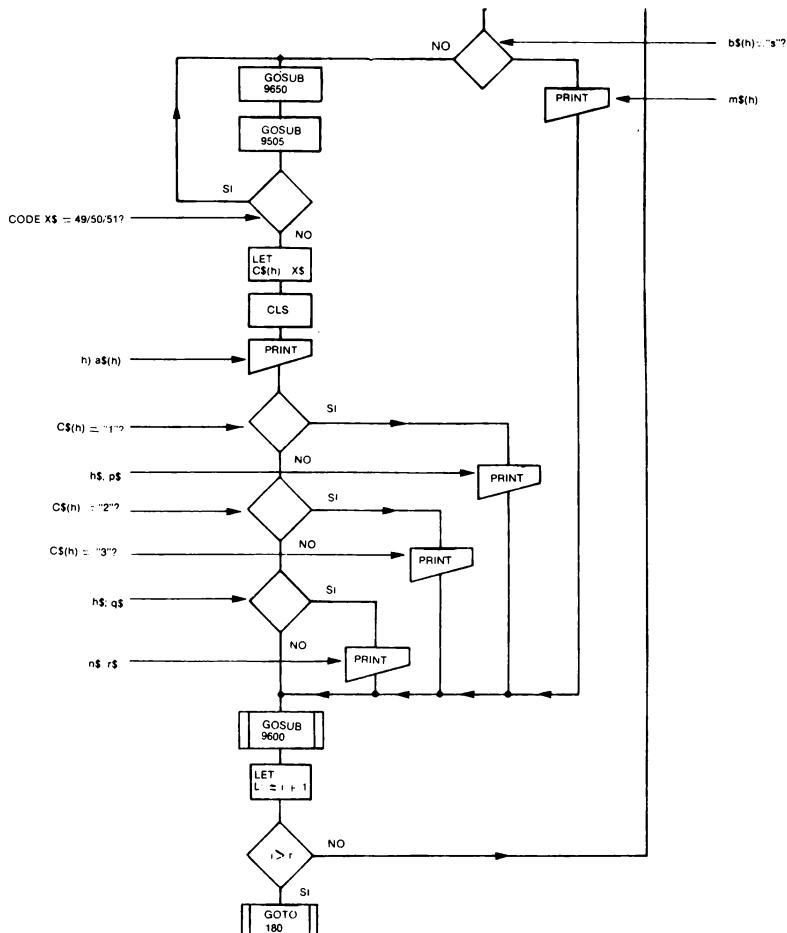
Sul metodo di lavoro da seguire ci è sembrato più semplice e immediato procedere all'inserimento contemporaneo dei dati e questo sulla base di due considerazioni:

- l'insegnante potrà meglio valutare le risposte inserite per ciascuna frase controllandole alla fine di ogni inserimento;
- introducendo le frasi con le relative risposte i tempi di esecuzione delle varie operazioni risulteranno dimezzati.

Si è infine ritenuta utile la visualizzazione, di volta in volta, delle frasi inserite e del giudizio espresso, questo per un immediato controllo dei dati inseriti.

Diagramma di flusso





Programma in BASIC (1103÷1910)

```

1103>FOR i=1 TO r
1105 FOR h=1 TO x
1107 IF CODE a$(h) = 32 THEN GO TO
1110
1108 NEXT h
1109 CLS : PRINT AT 10,6;"VERIFI
CA COMPLETA": GO TO 9550
1110 PRINT AT 10,1;"SCRIVI LA ";
h;" FRASE DA INSERIRE": PRINT FL
ASH 1; AT 21,28;"↑": GO SUB 9750
1120 INPUT a$(h)
1130 CLS
1135 INVERSE 1: PRINT TAB 0;h;"
";TAB 4;a$(h): INVERSE 0
1140 PRINT AT 10,3;m$;"?";TAB 13
;"(s/n)"
1145 GO SUB 9500
1155 LET b$(h) = x$

```

```

1157>CLS
1158 INVERSE 1: PRINT TAB 0;h;")
"TAB 4;a$(h): INVERSE 0
1159 IF b$(h)="s" THEN PRINT AT
10,5;m$: GO TO 1900
1160 GO SUB 9650
1170 GO SUB 9505
1172 IF CODE x$(>49 AND CODE x$(
>50 AND CODE x$(>51 THEN BEEP 1,
10: GO TO 1160
1190 LET c$(h)=x$
1200 CLS
1210 INVERSE 1: PRINT TAB 0;h;")
"TAB 4;a$(h): INVERSE 0
1220 IF c$(h)="1" THEN PRINT AT
10,2;n$;TAB 6;p$: GO TO 1900
1221 IF c$(h)="2" THEN PRINT AT
10,2;n$;TAB 2;q$: GO TO 1900
1222 IF c$(h)="3" THEN PRINT AT
10,2;n$;TAB 1;r$
1900 GO SUB 9600
1901 NEXT i
1910 GO TO 180

```

Il metodo di lavoro da noi scelto prevede la ripetizione delle stesse operazioni per ogni frase da inserire, pertanto la prima e la penultima linea di istruzioni dovranno chiudere un ciclo di ripetizione (ITERAZIONE).

Nella prima parte del sottoprogramma INSERIRE il numero delle frasi è definito dalla variabile z. L'iterazione conterà in i, incrementando di 1, il numero delle frasi sino al valore contenuto in r. Quando i risulterà superiore ad r il ciclo di ripetizione verrà interrotto dal NEXT i.

L'inserimento della frase è consentito dalla linea 1120 INPUT a\$(h); l'indice h numererà le frasi e ciò è indispensabile per l'accoppiamento delle risposte e per la ricerca dei dati, quando verrà usata la funzione CORREGGERE.

Una volta stampata la frase (linea 1135), l'insegnante dovrà specificarne le caratteristiche rispondendo alla prima domanda contenuta in m\$:

LA FRASE E' ACCETTABILE ?
(s/n)

La risposta verrà memorizzata in b\$(h) (linea 1155); se questa sarà affermativa b\$(h) ="s" si passerà all'inserimento della frase successiva, viceversa, l'insegnante dovrà indicare il motivo di non accettabilità:


```

                                PERCHE'
LA FRASE NON E' ACCETTABILE ?

SE DISORDINATA ----->PREMI 1
SE PER DISACCORDO
GRAMMATICALE ----->PREMI 2
SE PER DISACCORDO
DI SIGNIFICATO ----->PREMI 3

```

La nuova risposta sarà memorizzata in c\$(h) e la frase, completa delle sue caratteristiche, sarà visualizzata grazie alle linee di istruzione i da 1210 a 1900.

Il programma oltre a contenere istruzioni di controllo sui tasti da premere (linea 1172), presenta una particolare forma di controllo alle linee 1105,1107,1108.

Poiché le (x) frasi possono essere inserite di volta in volta, anche a distanza di tempo (quelle indicate con r dovranno invece essere inserite tutte), è necessario evitare di sovrapporre le nuove frasi a quelle già esistenti. I nuovi dati potranno accordarsi ai vecchi grazie alla linea 1107, che andrà alla ricerca del primo spazio vuoto nell'elenco delle frasi IF CODE a\$(h)=32.

La ricerca dello spazio dovrà avvenire per tutte le frasi già esistenti ed ecco la necessità della iterazione con indice h (es.: se sono state inserite 10 frasi, h avrà un valore 10; il primo spazio libero risulterà h=11 e la nuova frase sarà a\$(11)).

Nel caso l'elenco delle frasi risultasse già completo, l'istruzione 1109 visualizzerà la seguente scritta:

```

                                VERIFICA COMPLETA

                                PREMI UN TASTO PER CONTINUARE

```

e il sottoprogramma INSERIRE verrà interrotto grazie ad un'istruzione di salto contenuta nel GOSUB 9550 che rimanderà il programma al MENU PRINCIPALE.

Per controllare la verifica

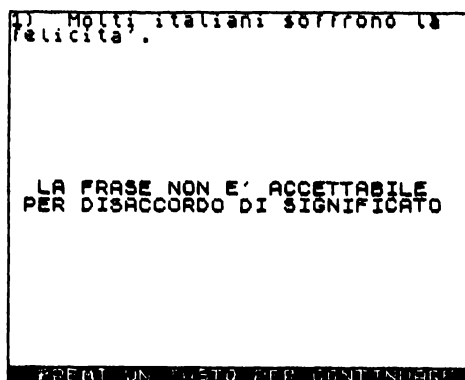
Nel caso l'insegnante voglia controllare i dati o correggere eventuali errori, da quelli ortografici a quelli contenutistici, potrà operare controlli e modifiche grazie alle seguenti funzioni:

LISTARE — CORREGGERE — STAMPARE

Queste funzioni risultano indispensabili per qualsiasi tipo di software.

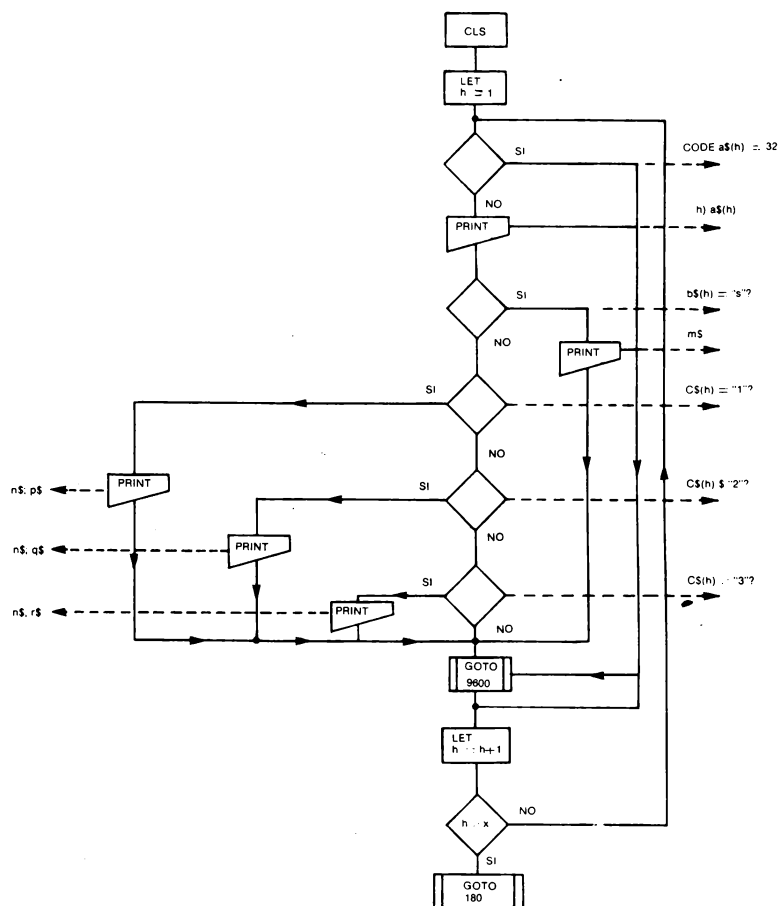
Listare

Premendo il tasto "5" comparirà sullo schermo l'elenco delle frasi inserite con le relative caratteristiche:



Poiché l'intero elenco non sarà contenuto nello spazio di una sola videata, l'insegnante potrà passare alle successive informazioni fino alla visualizzazione dell'ultimo dato, premendo ogni volta un tasto qualsiasi.

Diagramma di flusso



Programma in BASIC (5000÷5500)

```

5000>REM LISTARE
5005 CLS
5010 FOR h=1 TO x
5015 IF CODE a$(h)=32 THEN GO TO
5020 5130
5025 INVERSE 1: PRINT TAB 0;h;")
5030 TAB 4;a$(h): INVERSE 0: PRINT
5035 PRINT
5040 IF b$(h)="s" THEN PRINT AT
5045 4;m$: GO TO 5150
5050 IF c$(h)="1" THEN PRINT AT
5055 10;2;n$: TAB 6;p$: GO TO 5150
5060 IF c$(h)="2" THEN PRINT AT
5065 10;4;n$: TAB 2;q$: GO TO 5150
5070 IF c$(h)="3" THEN PRINT AT
5075 10;6;n$: TAB 1;r$:
5080 GO SUB 9500
5085 NEXT h
5090 GO TO 180

```

Poiché dovrà essere stampato l'intero elenco dei dati, frasi e caratteristiche di ognuna, si dovrà considerare un'iterazione all'inizio dell'intero programma BASIC.

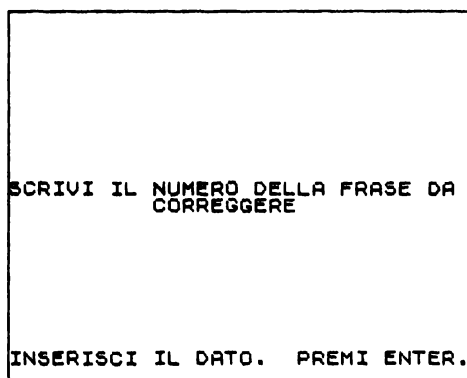
Le condizioni poste dall'IF... THEN, consentiranno la stampa delle caratteristiche, evitando qualsiasi tipo di errore.

La linea 5115 determinerà la fine dell'esecuzione del programma non appena il computer avrà individuato uno spazio vuoto della memoria a\$, quando cioè saranno esaurite le frasi da visualizzare.

Correggere

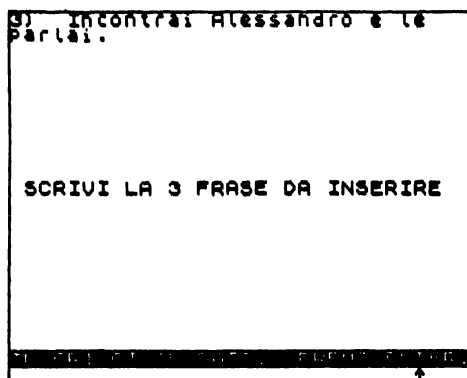
Come abbiamo già accennato questa funzione risulta indispensabile per rimediare ad eventuali errori di scrittura o di inserimento dati, sempre possibili durante la costruzione delle verifiche.

L'insegnante potrà sostituire nell'elenco quelle frasi a suo giudizio non adatte, perché ritenute non chiare o troppo complicate, dopo aver risposto alla seguente richiesta:



```
SCRIVI IL NUMERO DELLA FRASE DA  
CORREGGERE  
  
INSERISCI IL DATO.  PREMI ENTER.
```

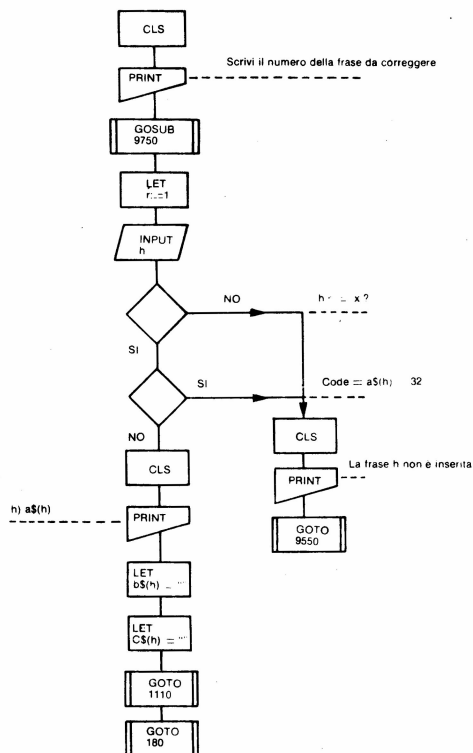
Indicata col numero dell'elenco la frase da correggere o da sostituire, questa verrà visualizzata sullo schermo:



```
9) Incontrai Alessandro e le  
parlai.  
  
SCRIVI LA 3 FRASE DA INSERIRE  
  
PREMI ENTER.
```

L'insegnante dovrà a questo punto riscrivere la frase apportando le dovute correzioni o, nel caso volesse sostituirla, inserire quella nuova.

Diagramma di flusso:



Programma i BASIC (2000÷2400)

```

2000 REM CORREGGERE
2100 CLS : PRINT AT 10,0;"SCRIVI
2110 IL NUMERO DELLA FRASE DA";TAB 1
2120 "CORREGGERE": GO SUB 9750
2130 LET F=1
2140 INPUT h
2150 IF h<=x THEN GO TO 2120
2160 CLS : PRINT AT 10,2;"LA FRA
2170 SE h;" NON E' INSERITA.": GO T
2180
2190 IF CODE a$(h)=32 THEN GO TO
2200
2210 CLS
2220 INVERSE 1: PRINT TAB 0;h;"
2230 TAB 4;a$(h): INVERSE 0
2240 LET b$(h)="" : LET c$(h)=""
2250 GO TO 1110
2260 GO TO 180
  
```

Come si può osservare il problema della correzione si traduce in un nuovo inserimento dei dati (GOTO 1110) e questo facilita la procedura, poiché non si dovrà operare alcun riordino dell'elenco.

Ai vecchi dati che saranno cancellati grazie alle istruzioni LET b\$(h)=" " e LET c\$(h)=" " si sostituiranno semplicemente i nuovi.

Le istruzioni che controllano un corretto modo di operare sono date dalle linee 2112 e 2115. Quest'ultima segnala che si vuole correggere una frase che non risulta in elenco.

La parola-chiave INVERSE sempre presente nei comandi di stampa della frase è stata utilizzata per evidenziare su fondo nero (INVERSE 1) la frase. Bisogna ricordarsi sempre di invertire nuovamente il video (INVERSE 0) per ritornare ad una scrittura su fondo bianco.

Stampare

Questo sottoprogramma va utilizzato per ottenere l'elenco delle frasi stampato su carta.

Sia il diagramma di flusso sia il programma in BASIC risulteranno simili a quelli del LISTARE con l'unica differenza dovuta alla parola-chiave LPRINT in sostituzione del comando PRINT.

Programma in BASIC (6000÷6500)

```
6000>REM STAMPARE
6100 CL0
6110 FOR h=1 TO X
6115 IF CODE a$(h)=32 THEN GO TO
6155
6120 INVERSE 1: LPRINT TAB 0;h;"
)";TAB 4;a$(h): INVERSE 0: LPRIN
T: LPRINT
6130 IF b$(h)="s" THEN LPRINT TA
B 4;"; GO TO 6150
6135 IF c$(h)="i" THEN LPRINT TA
B 2;";TAB 6;p$; GO TO 6150
6140 IF c$(h)="2" THEN LPRINT TA
B 2;";TAB 2;q$; GO TO 6150
6145 IF c$(h)="3" THEN LPRINT TA
B 2;";TAB 1;r$
6150 LPRINT: LPRINT
6155 NEXT h
6500 GO TO 100
```

Riportiamo di seguito l'elenco delle frasi da noi proposte per la verifica v f 1/1:

1) Molti italiani soffrono la
felicità.

LA FRASE NON E' ACCETTABILE
PER DISACCORDO DI SIGNIFICATO

2) Gli archeologi portarono al-
la luce numerosi reperti.

LA FRASE E' ACCETTABILE

3) Incontrai Alessandro e le parlai.

LA FRASE NON E' ACCETTABILE
PER DISACCORDO GRAMMATICALE

4) Quei giovani parlavano dell'ultimo film di Fellini.

LA FRASE E' ACCETTABILE

5) Il treno correva e i viaggiatori aspettavano che partisse.

LA FRASE NON E' ACCETTABILE
PER DISACCORDO DI SIGNIFICATO

6) Quell'automobilista a provocato una disgrazia.

LA FRASE NON E' ACCETTABILE
PER DISACCORDO GRAMMATICALE

7) Il gatto la bottiglia ha fatto di cadere latte.

LA FRASE NON E' ACCETTABILE
PERCHE' DISORDINATA

8) Alcuni aborigeni vivono nella foresta.

LA FRASE E' ACCETTABILE

9) I vigili sul prontamente dell'incendio accorsero luogo.

LA FRASE NON E' ACCETTABILE
PERCHE' DISORDINATA

10) Ho fame perche' vado a letto.

LA FRASE NON E' ACCETTABILE
PER DISACCORDO DI SIGNIFICATO

Registrare

Uno dei problemi che gli insegnanti, e il mondo della scuola in generale, da anni si trovano ad affrontare è dato dalla mancanza di circolazione delle esperienze didattiche già sperimentate e dalla assenza, nella maggior parte delle scuole, di un archivio dal quale attingere materiale e curricoli didattici da utilizzare nelle proprie attività.

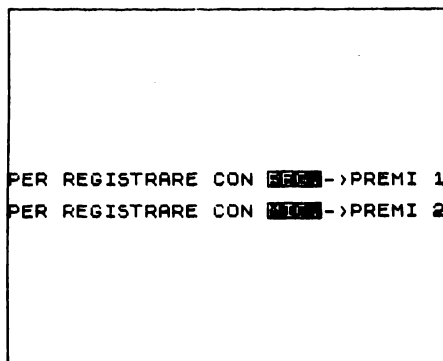
Nelle poche scuole dove si sono realizzati archivi di consultazione, la presenza di una quantità enorme di schede, pubblicazioni, unità didattiche, corsi di studio, ecc. crea non pochi problemi di ricerca. L'informatizzazione in questo caso risolverebbe radicalmente e definitivamente tale problema, riducendo drasticamente l'enorme quantità di carta e fogli e i tempi di ricerca da parte degli insegnanti.

Utilizzando delle comuni cassette magnetiche per registratori o “memorie di massa” più evolute quali floppy-disk o dischi rigidi, è possibile conservare qualsiasi tipo di dati.

Anche per la nostra verifica abbiamo considerato indispensabile sfruttare tali vantaggi e questo per alcuni motivi:

- poter utilizzare la prova per un numero quasi infinito di volte (il “quasi” dipende dall’usura dei materiali: cassette o dischi);
- consentire l’esecuzione della verifica, contemporaneamente, a più alunni;
- conservare in una software-biblioteca i lavori che si realizzano nel corso dell’attività didattica.

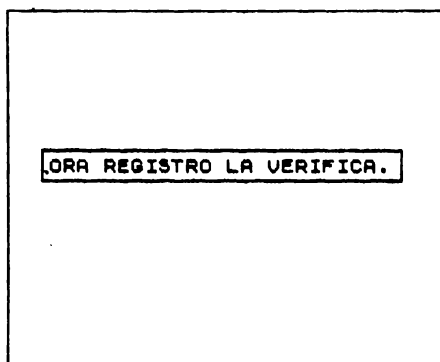
Premendo il tasto 9 comparirà sul video la seguente scritta:



Possedendo un registratore a cassette, collegabile al computer, basterà premere 1; viceversa, disponendo di un MICRODRIVE si premerà 2.

La registrazione del programma e dei dati in esso contenuti sarà direttamente gestita dalla macchina (nel caso si operi su SINCLAIR-ZX SPECTRUM e si voglia registrare su cassetta comparirà sul video la seguente richiesta: “start tape, then press any key” e cioè: FAI PARTIRE IL NASTRO E PREMI UN TASTO QUALSIASI).

Subito dopo la seguente comunicazione video il computer inizierà automaticamente la registrazione:



Programma in BASIC (9000÷9299)

```
9000>REM REGISTRARE
9005 CLS
9010 PRINT AT 10,0;"PER REGISTRA
RE CON REG.->PREMI 1"
9015 PRINT AT 12,0;"PER REGISTRA
RE CON MIC.->PREMI 2"
9020 GO SUB 9505
9022 IF X0="1" THEN GO TO 9100
9023 IF X0="2" THEN GO TO 9200
9024 BEEP 1,10: GO TO 9010
9100 SAVE "v/1/1" LINE 1
9105 GO TO 1
9200 ERASE "v/1/1"
9205 CLS
9210 PLOT 20,92: DRAW +210
9215 DRAW 0,+16: DRAW -210,0: DRA
U 0,-16: PRINT AT 9,3;"ORA REGIS
TRO LA VERIFICA."
9218 SAVE "v/1/1" LINE 1
9299 GO TO 1
```

Il programma sarà lo stesso per tutte le verifiche.

Le linee d'istruzione che consentono la registrazione su memorie di massa sono la 9100 e la 9200.

Senza entrare nel merito delle "formule" richieste dal sistema, precisiamo che è obbligatorio indicare il nome del programma in oggetto (nel nostro caso: v/1/1) ed è indispensabile salvare il programma con la parola-chiave LINE 1, per richiamarlo successivamente già in fase operativa ed impedire l'accesso alle linee del programma BASIC.

La registrazione della nuova verifica avverrà solo dopo aver cancellato dalla memoria di massa quella precedente. Ciò viene eseguito direttamente dal computer, grazie all'istruzione contenuta nella linea 9200.

Eeguire

La procedura da noi seguita per la realizzazione di tale sottoprogramma si suddivide nelle seguenti fasi:

- Azzeramento delle memorie;
- Memorizzazione dei dati dell'alunno;
- Esecuzione della prova;
- Stampa dei risultati della prova;
- Visualizzazione dei risultati della prova.

Lo schema a blocchi è dato in fig. 4.3.3.8

Per accedere a questo sottoprogramma, che consente l'esecuzione della prova, bisognerà premere il tasto 2 alla richiesta di riconoscimento.

Azzeramento delle memorie

Come è già stato detto, la verifica dovrà essere svolta da tutti gli alunni e per ognuno di essi dovranno, conseguentemente, essere registrati i dati e i risultati.

L'azzeramento delle memorie è necessario per predisporre il computer a ricevere i dati e le risposte che di volta in volta i diversi alunni dovranno inserire.

La prima comunicazione video sarà:

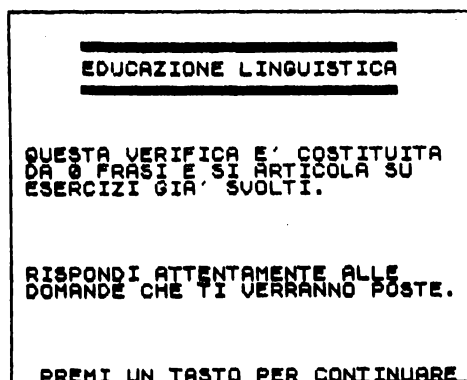
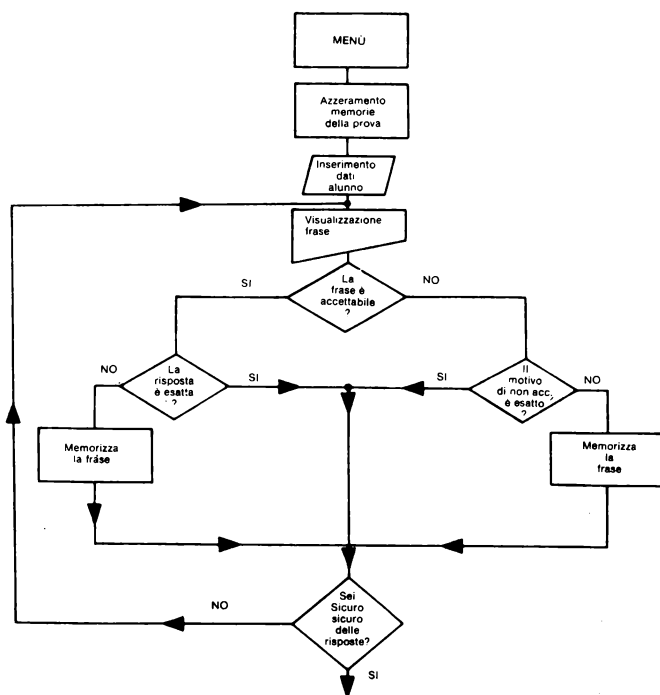
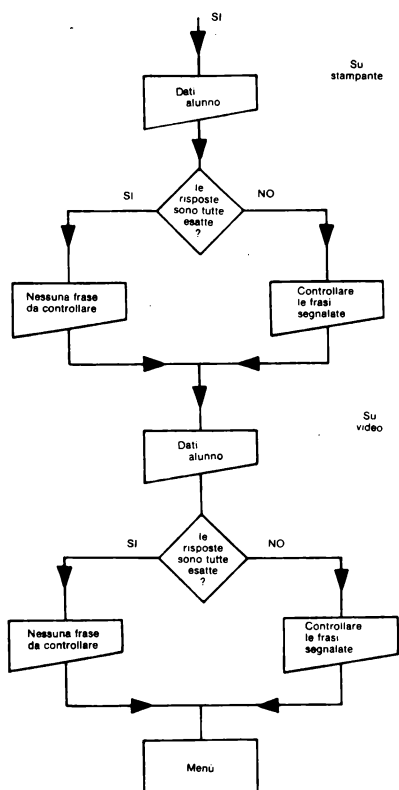


Diagramma a
blocchi ESEGUIRE





Programma in BASIC (8000÷8011)

```

8000>REM ESEGUIRE
8001 CLS
8002 FOR h=1 TO x
8003 IF CODE a$(h)=32 THEN GO TO
8005
8004 NEXT h
8005 LET s=h-1
8006 FOR h=1 TO x
8007 LET i$="": LET z$(h)="" : LE
T y$(h)=""
8008 NEXT h
8009 GO SUB 9710
8011 GO SUB 9600

```

L'iterazione contenuta nelle istruzioni 8002÷8004 permette di determinare il numero complessivo delle frasi inserite nella verifica.

La relazione presente all'interno dell'iterazione consente la numerazione corretta delle frasi, poiché in presenza di spazi liberi (CODE a\$(h)=32) il ciclo di ripetizione viene interrotto (GOTO 8005) e il valore assunto da h, diminuito di 1, sarà memorizzato in s.

Tale valore corrisponderà al numero delle frasi effettivamente presenti. Il comando che consente l'azzeramento degli indirizzi di memoria è ottenuto ponendo tutte le variabili, utilizzate dall'alunno, uguali a SPACE (" "). Tali istruzioni sono inserite nell'iterazione presenti nelle linee 8006÷8008 poiché dovranno essere cancellate le risposte dell'alunno precedente per tutte le frasi.

Memorizzazione dei dati dell'alunno

Questa fase consente la registrazione dei dati dell'alunno: cognome, nome, classe. Sullo schermo apparirà la prima richiesta.

SCRIVI IL TUO COGNOME E NOME

INSERISCI IL DATO. PREMI ENTER.

e successivamente:

SCRIVI LA CLASSE

INSERISCI IL DATO. PREMI ENTER.

Programma in BASIC (8013÷8036)

```
8013)PRINT AT 10,2;"SCRIVI IL TU  
O COGNOME E NOME";GO SUB 9750  
8015 INPUT i$(1 TO 24)  
8016 CLS  
8020 PRINT AT 10,7;"SCRIVI LA CL  
ASSE";GO SUB 9750  
8025 INPUT i$(26 TO 35)  
8036 CLS
```

L'osservazione da fare riguarda l'indirizzo di memoria predisposto a tale compito che risulta lo stesso per il cognome, il nome e la classe dell'alunno.

L'accorgimento adottato è detto SLICING. La memoria di indirizzo i\$ ha 35 caratteri a disposizione (vedi legenda variabili nel programma INIZIALIZZARE); di questi, 24 sono stati riservati ricevere le generalità dell'alunno i\$(1 TO 24); i rimanenti 10 per la classe di appartenenza i\$(26 TO 35).

Esecuzione della prova

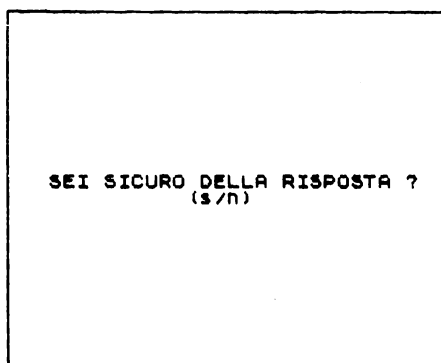
È dall'esecuzione di questa terza parte del sottoprogramma in oggetto che l'alunno comincerà a svolgere la prova di verifica.

Le frasi inserite dall'insegnante verranno riproposte all'allievo e per ognuna questi dovrà eseguire le stesse operazioni di analisi e di individuazione delle caratteristiche svolte dall'insegnante nel programma INSERIRE.

Visualizzata la frase, l'alunno sarà invitato a rispondere ai diversi quesiti e le risposte dell'allievo saranno memorizzate nelle DIM già predisposte durante la fase dell'INIZIALIZZARE.

Le comunicazioni video risulteranno le stesse utilizzate per l'insegnante nel programma INSERIRE FRASI.

Per ogni frase eseguita comparirà sul video la seguente scritta:



Se l'alunno sarà sicuro della risposta data premerà il tasto "S" e questa verrà memorizzata, viceversa premendo il tasto "N" la frase gli verrà riproposta offrendogli la possibilità di un nuovo inserimento.

Dal confronto delle risposte effettuato per ogni frase, si conosceranno i risultati della prova.

Programma in BASIC (8040÷8908)

Il programma che consente lo svolgimento della prova conserva la stessa struttura dell'INSERIRE, con alcune variazioni.

```
8040>FOR h=1 TO 5
8041 INVERSE 1: PRINT TAB 0;h;" )
      TAB 4;:INVERSE 0
8043 PRINT AT 10,3;:?" ;TAB 13
      ;(S/N)
8045 GO SUB 9500
```

```

8060>CLS
8065 IF x$="s" THEN GO TO 8700
8067 INVERSE 1: PRINT TAB 0;h;"
      TAB 4;a$(h): INVERSE 0
8070 GO SUB 9550
8075 GO SUB 9505
8080 IF CODE x$<>49 AND CODE x$<
      >50 AND CODE x$<>51 THEN BEEP 1,
      10: GO TO 8070
8100 IF x$=c$(h) THEN GO TO 8900
8119 LET y$(h)=x$
8120 LET z$(h)=a$(h)
8130 GO TO 8900
8700 IF x$=b$(h) THEN GO TO 8900
8705 LET z$(h)=a$(h)
8900 CLS: INVERSE 1: PRINT TAB
0;h;"      TAB 4;a$(h): INVERSE 0
8901 PRINT AT 10,3;"SEI SICURO D
      ELLA RISPOSTA ?";TAB 13;"($/n)"
8902 IF INKEY$<>"" THEN GO TO 89
82
8903 IF INKEY$="" THEN GO TO 890
3
8904 IF INKEY$="s" THEN GO TO 89
07
8905 LET z$(h)="" : LET y$(h)=""
8906 CLS : GO TO 8042
8907 CLS
8908 NEXT h

```

Come si può notare le istruzioni corrispondenti alle operazioni che l'utente dovrà eseguire, sono contenute in un ciclo di ripetizione (iterazione) perché per ogni frase dovrà avvenire il confronto con le risposte (linee 8065, 8100,8700).

La variabile del contatore h sarà s.

In caso di erroneo giudizio da parte dello studente, la frase in oggetto sarà memorizzata in z\$(h) e le risposte in y\$(h).

L'ultima linea di questa parte del programma ESEGUIRE si concluderà con la chiusura dell'iterazione: NEXT h

Stampa dei risultati della prova

Una volta eseguita la prova, il computer stamperà i risultati ottenuti dall'alunno e subito dopo li visualizzerà sullo schermo grazie alle istruzioni contenute nell'ultima parte del programma.

L'emissione dei dati sia su stampante che su video rappresenta una verifica dello stesso programma.

Questi, infatti, si potranno avere solo se il metodo di lavoro scelto sarà stato eseguito in modo logico e coerente e se il programma in BASIC risulterà completo in ogni sua parte.

Nel caso l'alunno abbia risposto esattamente su tutte le frasi verrà stampata e visualizzata la seguente scritta:

NESSUNA FRASE DA CONTROLLARE

viceversa:

```
ALUNNO:  
CLASSE:  
CONTROLLARE LE FRASI SEGNALATE:
```

con le frasi e le relative risposte.

L'insegnante e l'alunno potranno a questo punto, approfondire l'analisi dei dati e chiarire i dubbi sorti in fase di giudizio.

Programma in BASIC (8910÷8940)

```
8910 LPRINT AT 1,0:"ALUNNO:";:;:  
1 TO 24): LPRINT: LPRINT "CLAS  
SE:";:;:(25 TO 35)  
8911 FOR h=1 TO 5: IF CODE Z$(h)  
<>32 THEN GO TO 8916  
8912 NEXT h  
8913 LPRINT: LPRINT AT 5,0,"NES  
SUNA FRASE DA CONTROLLARE"  
8914 GO TO 8950  
8916 LPRINT: LPRINT AT 5,0,"CON  
TROLLARE LE FRASI SEGNALATE:"  
8918 LPRINT: LPRINT  
8920 FOR h=1 TO 5  
8921 IF CODE Z$(h)=32 THEN GO TO  
8940  
8922 INVERSE 1: LPRINT TAB 0,h:"  
";:TAB 4,Z$(h): INVERSE 0: LPRIN  
T: LPRINT  
8923 IF CODE Z$(h)<>32 AND CODE  
Y$(h)<>32 THEN LPRINT "RISPOSTA:"  
;:n$;  
8924 IF CODE Z$(h)<>32 AND CODE  
Y$(h)=32 THEN LPRINT "RISPOSTA:"  
;:n$;  
8925 IF CODE Z$(h)=32 AND CODE Y  
$(h)<>32 THEN LPRINT "RISPOSTA:"  
;:n$;  
8926 IF Y$(h)="1" THEN LPRINT P$  
8929 IF Y$(h)="2" THEN LPRINT Q$  
8930 IF Y$(h)="3" THEN LPRINT R$  
8931 LPRINT: LPRINT  
8940 NEXT h
```

La condizione che consente l'esecuzione di questa parte del programma ESEGUIRE è data dalla linea 8911.

Le istruzioni presenti controlleranno se sono state memorizzate frasi in Z\$(h); in questo caso, essendo stato commesso un errore di valutazione da parte dello studente, la macchina darà in OUTPUT la frase con le relative risposte (8916÷8940).

In caso contrario se non sono stati registrati errori, verrà interrotto alla linea 8914 che rimanda al 8950 ovvero alla fase di VISUALIZZAZIONE DEI RISULTATI DELLA PROVA.

Visualizzazione dei risultati della prova

Per quest'ultima parte del programma valgono le considerazioni fatte per la stampa; il programma in BASIC risulterà diverso solo per il comando PRINT al posto di LPRINT.

Programma in BASIC (8950÷8991)

```
8950>PRINT AT 1,0;"ALUNNO: ";i$(
1 TO 24)
8952 PRINT : PRINT "CLASSE: ";i$
(25 TO 35)
8953 FOR h=1 TO s: IF CODE z$(h)
<>32 THEN GO TO 8956
8954 NEXT h
8955 PRINT AT 5,0;".. NESSUNA FRA
SE DA CONTROLLARE
8956 GO TO 8990
8958 PRINT AT 5,0;" CONTROLLARE
LE FRASI SEGNALATE:"
8959 PRINT : PRINT
8960 FOR h=1 TO s
8961 IF CODE z$(h)=32 AND CODE y
$(h)=32 THEN GO TO 8977
8962 INVERSE 1: PRINT TAB 0;h;"
":TAB 4;z$(h): INVERSE 0: PRINT
: PRINT
8963 IF CODE z$(h)<>32 AND CODE
y$(h)<>32 THEN PRINT "RISPOSTA:"
n$;
8964 IF CODE z$(h)<>32 AND CODE
y$(h)=32 THEN PRINT "RISPOSTA:"
n$;
8965 IF CODE z$(h)=32 AND CODE y
$(h)<>32 THEN PRINT "RISPOSTA:"
n$;
8966 IF y$(h)="1" THEN PRINT p$
8969 IF y$(h)="2" THEN PRINT q$
8970 IF y$(h)="3" THEN PRINT r$
8971 PRINT : PRINT
8977 NEXT h
8990 GO SUB 9800
8991 GO TO 1
```

4.3.4 vf 1/2 - Regole di accordo

I GOSUB

I GOSUB presenti in questa verifica sono già stati illustrati precedentemente. Riportiamo quindi solo quelli peculiari della vf 1/2.

GOSUB 9800

Programmi in BASIC (9800÷9805)

```
9800>REM GO SUB
9801 INVERSE 1: PRINT TAB 0;h;"
":TAB 4;s$(h): PRINT : PRINT TAB
0;"SOLUZIONI:" INVERSE 0: PRIN
T
9802 FOR t=1 TO p: PRINT TAB 2;t
;"":TAB 4;b$(h,t);TAB 4;c$(h,t)
: PRINT : NEXT t
9805 RETURN
```


Queste istruzioni consentono di visualizzare:

- la frase (linea 9801);
- il numero e le possibili soluzioni della frase, il termine con cui si accorda l'elemento completato (linea 9802) stabiliti dall'insegnante durante la fase INSERIMENTO FRASI.

GOSUB 9850

Programma in BASIC (9850÷9855)

```
9850>REM GO SUB
9851 INVERSE 1: PRINT TAB 0;h;")
"TAB 4;e8(h);: PRINT : PRINT TAB
0;"SOLUZIONI:": INVERSE 0: PRIN
T
9852 FOR t=1 TO 0: PRINT TAB 2;t
;";TAB 4;d8(h,t);TAB 4;e8(h,t)
: PRINT : NEXT t
9855 RETURN
```

Come si può notare questo GOSUB risulta simile nella sua struttura a quello precedente. In questo caso però verranno visualizzate le risposte date dall'alunno durante la fase ESEGUIRE.

Menù

Il programma MENU si compone di tre parti:

- presentazione verifica;
- codice d'accesso;
- menù insegnanti.

Presentazione verifica

La verifica inizia con una comunicazione video che si presenta nel seguente modo:

EDUCAZIONE LINGUISTICA

OBIETTIVO: saper valutare
l'importanza comunicativa delle
regole di accordo.

Questa verifica rappresenta il
momento conclusivo del lavoro
precedentemente svolto in clas-
se su:
LE REGOLE DI ACCORDO

PREMI UN TASTO PER INIZIARE

Programma in BASIC (1÷50)

```
1>REM MENU
2 CLS : BORDER 1: PAPER 7: IN
K 0: CLS
3 PRINT AT 2,5; PAPER 6; "
5: PAPER 6;"EDUCAZIONE LINGUISTI
CA"
```

```

20 PRINT AT 4,5;"PAPER 6;"
30 PRINT AT 7,1;"OBIETTIVO:
saper valutare l'importanza
comunicativa del- le regole di
accordo."
40 PRINT AT 12,1;"Questa verif
ica rappresenta il momento conc
lusivo del lavoro precedente me
nte svolto in clas- se su:";TAB
5;"LE REGOLE DI ACCORDO"
50 PRINT AT 20,3;"PREMI UN TAS
TO PER INIZIARE": PAUSE 0: GO TO
60

```

Codice d'accesso

Programma in BASIC (55÷170)

```

55>PRINT AT 20,7; PAPER 6;"COD
ICE SCONOSCIUTO": PAUSE 150
60 PAUSE 100: CLS: PLOT 5,75:
DRAW PAPER 6,+245,0: DRAW PAPER
6,0,-40: DRAW PAPER 6,-245,0: D
RAW PAPER 6,0,+40
70 PLOT 5,99: DRAW PAPER 6,+24
5,0: DRAW PAPER 6,0,+40: DRAW PA
PER 6,-245,0: DRAW PAPER 6,0,-40
75 PRINT AT 7,2;"PER GLI INSEG
NANTI-----": PAPER 4;"INSEG
60 PRINT AT 15,2;"PER GLI ALUN
NI-----": PAPER 2; INK 7
;"2"
85 PRINT AT 1,4;"PER INIZIARE
LA VERIFICA": PAINT TAB 7;"FARSI
RICONOSCERE"
90 GO SUB 9505
100 IF CODE X$=49 THEN GO TO 15
0
110 IF CODE X$=50 THEN GO TO 00
00
120 IF CODE X$(<>49 AND CODE X$(<
50 THEN BEEP 1,10: GO TO 90
150 CLS: PRINT AT 10,3: FLASH
1;"SCRIVI IL CODICE D'ACCESSO"
160 GO SUB 9505
165 IF CODE X$=46 THEN GO TO 10
0
170 IF CODE X$(<>46 THEN BEEP 1,
10: GO TO 55

```

Menù insegnanti (menù principale)

Sul video comparirà il seguente menù:

EDUCAZIONE LINGUISTICA VERIFICA REGOLE DI ACCORDO	PER CONTROLLARE LA VERIFICA LISTARE-----5 CORREGGERE----2 STAMPARE-----6
PER COSTRUIRE LA VERIFICA INIZIALIZZARE-----4 INSERIRE FRASI-----1	PER REGISTRARE LA VERIFICA---9 COMPLETATE LE OPERAZIONI PREMI-----0

Anche le fasi di lavoro saranno le medesime:

- ## Programma in BASIC (180÷510)

Per costruire la verifica

- INIZIALIZZARE;
- INSERIRE FRASI.

- dimensionamento della verifica;
- individuazione e dimensionamento delle variabili.

a. Dimensionamento della verifica

Programma in BASIC (4000÷4080)

```
4000>REM INIZIALIZZARE
4010 CLS : PRINT AT 10,3:"VUOI C
ANCELLARE LE FRASI ?";TAB 13;"(S
/N)"
4020 GO SUB 9500
4040 IF X$="" THEN GO TO 4060
4050 GO TO 100
4060 CLS
4070 PRINT AT 10,2:"QUANTE FRASI
DEVE CONTENERE";TAB 9;"LA VERIF
ICA ?": GO SUB 9750
4080 INPUT X
```

b. Individuazione e dimensionamento delle variabili

Poiché la verifica su le REGOLE DI ACCORDO si basa sui significati che una frase può assumere in seguito al completamento di un suo termine e all'accordo di questo con un altro elemento della frase, nell'individuazione delle variabili sarà opportuno definire gli obiettivi che ci proponiamo:

- conoscere il numero di soluzioni possibili per il termine da completare
- conoscere per ogni soluzione come il termine è stato completato;
- conoscere per ogni termine completato l'elemento che si accorda;
- conoscere gli eventuali errori commessi dall'alunno durante la verifica.

Per i primi tre obiettivi bisognerà considerare i dati espressi sia dall'insegnante sia dall'alunno.

Legenda delle variabili

VARIABILE	FUNZIONE	DIMENSIONE DELLA MEMORIA
a\$(x, 92)	vedi a\$ di vf 1/1	vedi a\$ vf1/1
b\$(x,4,14)	conservierà il termine completato dall'insegnante	x:=n° max. di frasi della verifica 4:=n° max. di soluzioni previste 14:=n° max. di caratteri disponibili per il termine da completare
c\$(x,4,14)	conservierà l'elemento scelto dall'insegnante che si accorda con il termine completato	x:=n° max. di frasi della verifica 4:=n° max. di soluzioni previste 14:=n° max. di caratteri disponibili per per l'elemento che si accorda con il termine completato
d\$(x,4,14)	conservierà il termine completato dall'alunno	vedi b\$
l\$(x,14,14)	conservierà l'elemento scelto dall'alunno che si accorda con il termine completato	vedi c\$
y\$(x,4,14)	conservierà il termine completato dall'alunno ordinandolo secondo la sequenza operata dall'insegnante	vedi d\$
z\$(x,4,14)	conservierà l'elemento scelto dall'alunno che si accorda con il termine completato ordinandolo secondo le sequenze operata dall'insegnante	vedi e\$

VARIABILE	FUNZIONE	DIMENSIONE DELLA MEMORIA
w\$(x,92)	vedi z\$ di vf1/1	vedi z\$ di vf 1/1
p\$(x,1)	conserverà il numero di soluzioni stabilite dall'insegnante per il termine da completare	x:=n° max. di frasi della verifica l:=n° carattere disponibile
O\$(x,1)	conserverà il numero di soluzioni stabilito dall'alunno per il termine da completare	vedi p\$
i\$(35)	vedi i. di vf1/1	vedi i\$ di vf 1/1

Programma in BASIC (4200÷4900)

```

4200>DIM a$(x,92)
4201 DIM b$(x,4,14)
4202 DIM c$(x,4,14)
4203 DIM d$(x,4,14)
4204 DIM e$(x,4,14)
4210 DIM u$(x,4,14)
4211 DIM z$(x,4,14)
4212 DIM w$(x,92)
4213 DIM p$(x,1)
4214 DIM o$(x,1)
4215 DIM i$(35)
4290 CLS
4300 LET s$=" PREMI UN TASTO PE
R CONTINUARE "
4590 GO SUB 9700
4900 GO TO 9550

```

Inserire frasi

Il programma si suddivide, come per la verifica vf 1/1, in:
a) determinazione del numero delle frasi da inserire;
b) inserimento frasi.

Determinazione del numero delle frasi da inserire

Programma in BASIC (1000÷1102)

```

1000>REM INSERIRE
1010 CLS
1050 GO SUB 9700
1051 GO SUB 9600
1100 PRINT AT 10,2;"QUANTE FRASI
VUOI INSERIRE ?"
1101 GO SUB 9750

```

b) Inserimento frasi

Determinato il numero delle frasi l'insegnante potrà procedere all'inserimento di queste e di tutti gli elementi definiti nella fase di individuazione e dimensionamento delle variabili.

Le frasi da noi considerate offrivano al massimo 3 possibili tipi di soluzione, viceversa il programma realizzato prevede una quarta possibilità, questo in relazione al tipo di frase che l'insegnante potrà scegliere. L'inserimento della frase dovrà essere effettuato lasciando incompleta solo la parte finale del termine prescelto.

Inserita la frase, sul video comparirà la seguente comunicazione:

```
1) Ho classificato con Marco la  
foto che h... fatto.  
  
QUANTE SOLUZIONI HA LA FRASE?
```

Per il numero di soluzioni indicate, l'insegnante dovrà compiere le seguenti operazioni:

```
1) Ho classificato con Marco la  
foto che h... fatto.  
  
SCRIVI LA 1 SOLUZIONE  
INSERISCI IL DATO. PREMI ENTER.
```

```
1) Ho classificato con Marco la  
foto che h... fatto.  
  
SOLUZIONI:  
1) ha  
2)  
  
SCRIVI L'ELEMENTO DELLA FRASE  
CHE SI ACCORDA CON LA  
1 SOLUZIONE  
INSERISCI IL DATO. PREMI ENTER.
```

Nel caso di 2 soluzioni ecco la sequenza video:

```

1) Ho classificato con Marco le
foto che h... fatto.

SOLUZIONI:

1) ha
   Marco

2)

SCRIVI LA 2 SOLUZIONE
INSERISCI IL DATO. PREMI ENTER.

```

```

1) Ho classificato con Marco le
foto che h... fatto.

SOLUZIONI:

1) ha
   Marco

2) ho

SCRIVI L'ELEMENTO DELLA FRASE
CHE SI ACCORDA CON LA
2 SOLUZIONE
INSERISCI IL DATO. PREMI ENTER.

```

Completate le operazioni i dati saranno così visualizzati:

```

1) Ho classificato con Marco le
foto che h... fatto.

SOLUZIONI:

1) ha
   Marco

2) ho
   io

PREMI UN TASTO PER CONTINUARE

```

Risulterà non corretta la soluzione “hanno” perché il termine di accordo “loro” non è presente nella frase.

Il principio su cui si basa la prova, e di conseguenza su cui si baseranno i confronti tra le soluzioni e gli accordi definiti dall'insegnante e quelli indicati dall'alunno, è che ad ogni soluzione debba corrispondere un solo termine presente e visivamente individuabile fra quelli costituenti la frase.

Programma in BASIC (1103÷1999).

```
1103>FOR i=1 TO r
1105 FOR h=1 TO x
1107 IF CODE a$(h)=32 THEN GO TO
1110
1108 NEXT h
1109 CLS : PRINT AT 10,6;"VERIFI
CA COMPLETA": GO TO 9550
1110 PRINT AT 10,1;"SCRIVI LA "
h;"FRASE DA INSERIRE": PRINT AT
21,26;"↑": GO SUB 9750
1120 INPUT a$(h)
1130 CLS
1135 PRINT TAB 0; PAPER 6;h;"";
TAB 4;a$(h)
1140 PRINT AT 10,2;"QUANTE SOLUZ
IONI HA LA FRASE?"
1143 GO SUB 9505
1145 IF CODE x$(48 OR CODE x$>52
THEN BEEP 1,10: GO TO 1140
1146 LET p$(h)=x$
1147 LET p=VAL p$(h)
1148 CLS
1150>FOR j=1 TO p
1155 PRINT AT 0,0; PAPER 6;h;""
TAB 4;a$(h); PRINT AT 10,6;"SCR
IVI LA "j;"SOLUZIONE"
1158 GO SUB 9750
1160 INPUT b$(h,j)
1162 CLS
1165 GO SUB 9800
1170 PRINT AT 17,0;"SCRIVI L'EL
EMENTO DELLA FRASE";TAB 6;"CHE S
I ACCORDA CON LA ";TAB 10;j;"SO
LUZIONE"
1175 GO SUB 9750
1180 INPUT c$(h,j)
1182 CLS
1185 GO SUB 9800
1250 NEXT j
1890 GO SUB 9500
1900 NEXT i
1999 GO TO 180
```

Dei seguenti sottoprogrammi LISTARE-CORREGGERE-STAMPARE riportiamo solo i listati per mostrare al lettore gli aspetti peculiari della presente verifica. L'impostazione dei programmi è la stessa utilizzata per i corrispondenti programmi della vf 1/1

Listare

Programma in BASIC (5000÷5190)

```
5000>REM LISTARE
5100 CLS
5110 FOR h=1 TO x
5120 IF CODE a$(h)=32 THEN GO TO
5130
5130 LET p=VAL p$(h)
5140 INVERSE 1: PRINT TAB 0;h;""
TAB 4;a$(h); PRINT : PRINT TAB
0;"SOLUZIONI": INVERSE 0: PRIN
T
5150 FOR j=1 TO p
5160 PRINT TAB 2;j;"";TAB 4;b$(
h,j);TAB 4;c$(h,j): PRINT
5170 NEXT j
5175 GO SUB 9500
5180 NEXT h
5190 GO TO 180
```

Correggere

Programma in BASIC (2000÷2400)


```

2000>REM CORREGGERE
2100 CLS : PRINT AT 10,0;"SCRIVI
IL NUMERO DELLA FRASE DA":TAB 1
0;"CORREGGERE.": GO SUB 9750
2105 LET r=1
2110 INPUT h
2112 IF h<=x THEN GO TO 2120
2115 CLS : PRINT AT 10,2;"LA FRA
SE ":"h;" NON E' INSERITA.": GO T
O 9550
2120 IF CODE a$(h)=32 THEN GO TO
2115
2125 CLS
2130 INVERSE 1: PRINT TAB 0;h;"
":TAB 4;a$(h): INVERSE 0
2135 FOR j=1 TO 4: LET b$(h,j)="
":LET c$(h,j)="": NEXT j
2140 GO TO 1110
2400 GO TO 100

```

Stampare

Programma in BASIC (6000÷6190)

```

6000>REM STAMPARE
6100 CLS
6110 FOR h=1 TO x
6120 IF CODE a$(h)=32 THEN GO TO
6150
6130 LET p=VAL p$(h)
6140 INVERSE 1: LPRINT TAB 0;h;"
":TAB 4;a$(h): LPRINT : LPRINT
TAB 0;"SOLUZIONI.": INVERSE 0: L
PRINT
6150 FOR j=1 TO p
6160 LPRINT TAB 2;j;"":TAB 4;b$(
h,j):TAB 4;c$(h,j): LPRINT
6170 NEXT j
6175 LPRINT
6180 NEXT h
6190 GO TO 100

```

Riportiamo di seguito l'elenco delle frasi da noi proposte per la verifica vf 1/2

1) Ho classificato con Marco le
foto che h... fatto.

SOLUZIONI:

- 1) ha
Marco
- 2) ho
io

2) Ho detto a Franco che dev...
studiare.

SOLUZIONI:

- 1) devò
io
- 2) deve
Franco

3) Ho detto a loro che dovr...
partire sabato per Parigi.

SOLUZIONI:

1) dovro'
io

2) dovremo
noi

3) dovranno
loro

4) Ho visto un film con Paola
che era tristissim...

SOLUZIONI:

1) tristissimo
film

2) tristissima
Paola

5) Il comandante annuncio' ai
soldati che dovev... allontanar-
si dal campo.

SOLUZIONI:

1) doveva
comandante

2) dovevano
soldati

6) Ho chiesto a loro se poss...
ascoltare la relazione.

SOLUZIONI:

1) posso
io

2) possiamo
noi

3) possono
loro

7) Carlo ha detto agli amici di
essere piu' sincer...

SOLUZIONI:

1) sincero
Carlo

2) sinceri
amici

8) Ho detto agli alunni che do-
vev... controllare le verifiche.

SOLUZIONI:

1) dovevo
io

2) dovevano
alunni

3) dovevamo
noi

9) Barbara ha comperato un li-
bro all'amico carissim...

SOLUZIONI:

- 1) carissimo
amico

10) L'autista disse ai passeggeri che dovev... scendere per consegnare i documenti.

SOLUZIONI:

- 1) doveva
autista
- 2) dovevano
passeggeri

Registrare

Programma in BASIC (9000÷9299)

```
9000>REM REGISTRARE
9005 CLS
9010 PRINT AT 10,0;"PER REGISTRA
RE CON REG.->PREMI 1"
9015 PRINT AT 12,0;"PER REGISTRA
RE CON MIC.->PREMI 2"
9020 GO SUB 9505
9022 IF X$="1" THEN GO TO 9100
9023 IF X$="2" THEN GO TO 9200
9024 BEEP 1,10: GO TO 9010
9100 SAVE "Vf1/2" LINE 1
9105 GO TO 1
9200 ERASE "A":1;"Vf1/2"
9205 CLS: PLOT 20,92: DRAW +210
,0: DRAW 0,+15: DRAW -210,0: DRA
W 0,-15: PRINT AT 9,3;"ORA REGIS
TRO LA VERIFICA."
9210 SAVE "A":1;"Vf1/2" LINE 1
9299 GO TO 1
```

Eseguire

Le frasi operative del programma ESEGUIRE risulteranno le stesse della vf 1/1:

- azzeramento delle memorie;
- memorizzazione dei dati dell'alunno;
- esecuzione della prova;
- stampa dei risultati della prova;
- visualizzazione dei risultati della prova.

Azzeramento delle memorie

Questa fase, che ha la funzione di azzerare tutte le memorie utilizzate direttamente dall'alunno (d\$;e\$) e quelle predisposte alla memorizzazione degli errori (w\$;y\$;z\$), è caratterizzata da una doppia iterazione (h,j).

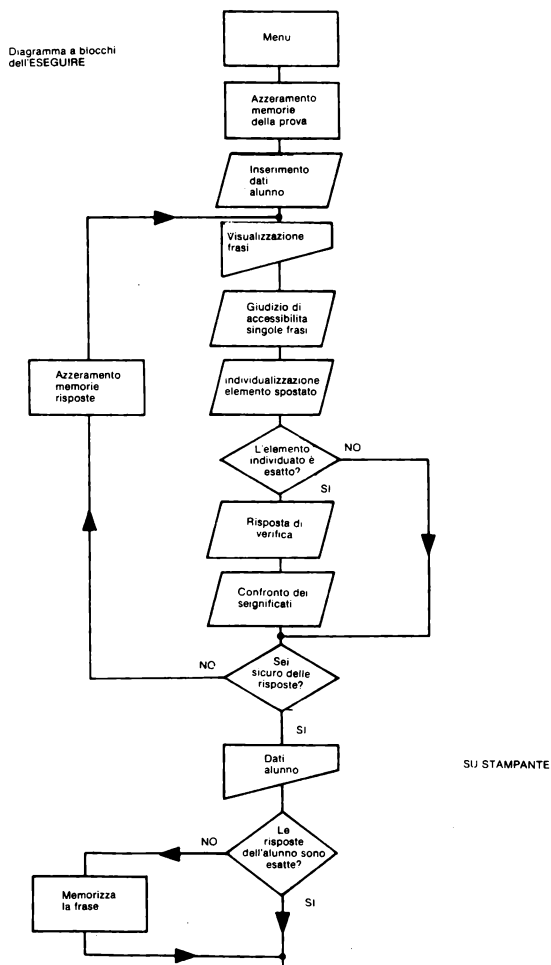
Programma in BASIC (8000-8011).

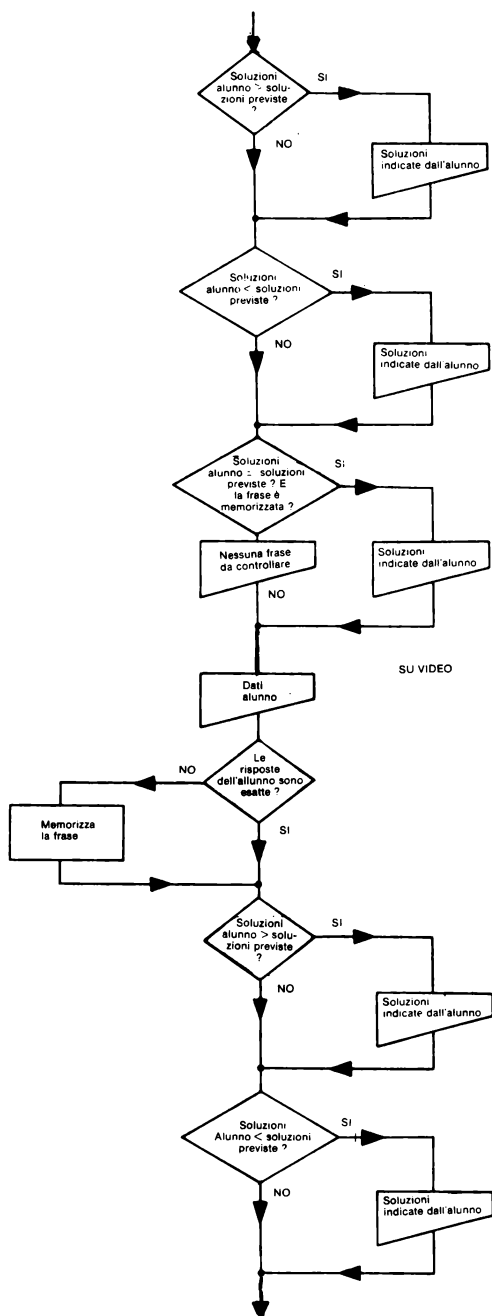
```

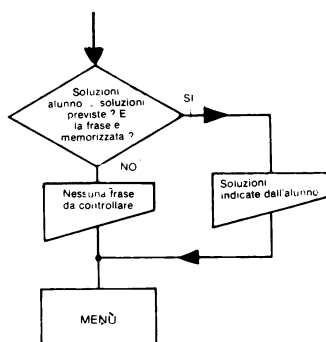
8000>REM ESEGUIRE
8001 CLS
8002 FOR h=1 TO x
8003 IF CODE a$(h)=32 THEN GO TO
8005
8004 NEXT h
8005 LET s=h-1
8006 FOR h=1 TO x
8007 FOR j=1 TO 4: LET d$(h,j)="
": LET e$(h,j)="": LET u$(h,j)="":
LET y$(h,j)="": LET z$(h,j)="":
LET i$="": NEXT j
8008 NEXT h
8010 GO SUB 9710
8011 GO SUB 9600

```

Le linee di istruzione del programma in BASIC sono le stesse utilizzate per la vf 1/1; le uniche variazioni sono costituite dai diversi indirizzi di memoria definiti per la prova.







Memorizzazione dei dati dell'alunno

Programma in BASIC (8013÷8036)

```

8013>PRINT AT 10,2;"SCRIVI IL TU
O COGNOME E NOME"; GO SUB 9750
8015 INPUT i$(1 TO 24)
8016 CLS
8020 PRINT AT 10,7;"SCRIVI LA CL
ASSE"; GO SUB 9750
8025 INPUT i$(26 TO 35)
8036 CLS
  
```

Esecuzione della prova

Visualizzata la frase, l'alunno dovrà indicare il numero delle soluzioni da lui individuate. Sulla base di queste, l'alunno dovrà compiere le stesse operazioni di inserimento dei dati già effettuate dall'insegnante durante la fase INSERIMENTO FRASI.

Dal confronto delle risposte e delle soluzioni indicate si conosceranno i risultati della verifica.

Programma in Basic (8040÷8511)

```

8040>FOR h=1 TO 5
8042 CLS : INVERSE 1: PRINT TAB
0,h;" ";TAB 4;a$(h); INVERSE 0
8050 PRINT AT 10,2;"QUANTE SOLU
ZIONI HA LA FRASE?"
8060 GO SUB 9505
8062 IF CODE X$(48 OR CODE X$>52
 THEN BEEP 1,10: GO TO 8050
8065 LET o$(h)=X$
8067 LET o=VAL o$(h)
8070 CLS
8080 FOR k=1 TO o
8090 INVERSE 1: PRINT AT 0,0,h;"
";TAB 4;a$(h); INVERSE 0: PRINT
AT 18,5;"SCRIVI LA ";k;" SOLUZI
ONE"
8100 GO SUB 9750
8110 INPUT d$(h,k)
8115 CLS
8150 GO SUB 9850
8220>PRINT AT 17,0;" SCRIVI L'EL
EMENTO DELLA FRASE";TAB 5;"CHE S
I ACCORDA CON LA ";TAB 10;k;" SO
LUZIONE"
8230 GO SUB 9750
8240 INPUT e$(h,k)
8241 CLS
8260 GO SUB 9850
8500 NEXT k
  
```

```

8502 PRINT AT 19,3;"SEI SICURO D
ELLE RISPOSTE ?";TAB 13;"($/n)"
8504 GO SUB 9500
8505 IF x$="s" THEN GO TO 8510
8507 FOR j=1 TO 4: LET d$(h,j)="
: LET e$(h,j)="": NEXT j
8508 CLS : GO TO 8042
8510 CLS
8511 NEXT h

```

Stampa dei risultati della prova

L'importanza di questa parte del sottoprogramma ESEGUIRE è legata al tipo di informazioni che desideriamo conoscere alla fine della prova sostenuta dall'alunno. Il programma da noi realizzato consente di ottenere:

- le frasi contenenti risposte errate;
- le soluzioni sbagliate dall'alunno;
- gli accordi erroneamente segnalati.

Inoltre, sarà possibile sapere se il numero delle soluzioni indicato dall'alunno risulta maggiore-uguale-minore rispetto a quello previsto.

La difficoltà da noi incontrata riguardava la presenza di diversi dati per una sola frase.

L'alunno infatti non conoscendo l'ordine delle risposte date dall'insegnante, inserirà le sue in modo arbitrario. È stato necessario quindi operare un confronto tra ogni singola risposta dell'allievo, con tutte quelle inserite dall'insegnante. Ciò è stato possibile grazie ad una procedura di controllo che, all'interno di un'iterazione il cui codice è determinato dal numero delle soluzioni indicato dall'allievo, memorizzerà in DIM diverse solo quelle che risulteranno sbagliate.

Programma in BASIC (8800-8894)

```

8800 LPRINT AT 1,0;"ALUNNO: ";i$
(1 TO 24): LPRINT : LPRINT "CLAS
SE : ";i$(26 TO 35): LPRINT
8810 FOR h=1 TO s
8815 LET p=VAL p$(h): LET o=VAL
o$(h)
8830 FOR j=1 TO p
8835 FOR t=1 TO o
8840 IF b$(h,j)=d$(h,t) AND c$(h,
j)=e$(h,t) THEN LET y$(h,t)=d$(
h,t): LET z$(h,t)=e$(h,t): LET d
$(h,t)=d$(h,j): LET e$(h,t)=e$(h,
j): LET d$(h,j)=y$(h,t): LET e$(
h,j)=z$(h,t): GO TO 8852
8842 NEXT t
8845 LET w$(h)=a$(h)
8852 NEXT j
8855 IF p<o THEN INVERSE 1: LPRIN
T TAB 0;h;"":TAB 4;a$(h): LPRIN
T : LPRINT TAB 0;"SOLUZIONI:"
INVERSE 0: LPRINT : LPRINT "L'AL
UNNO HA INDICATO UN NUMERO DI S
OLUZIONI SUPERIORE": LPRINT : LP
RINT TAB 0: PAPER 6:"SOLUZIONI",
"RISPOSTE ALUNNO": FOR t=1 TO o:
LPRINT : LPRINT TAB 2,t;"":TAB
4;b$(h,t);d$(h,t);TAB 4;c$(h,t)
;e$(h,t): LPRINT : NEXT t: GO TO
8890
8860 IF p>o THEN INVERSE 1: LPRIN
T TAB 0;h;"":TAB 4;a$(h): LPRIN
T : LPRINT TAB 0;"SOLUZIONI:"
INVERSE 0: LPRINT : LPRINT "L'AL
UNNO HA INDICATO UN NUMERO DI S
OLUZIONI INFERIORE": LPRINT : LP
RINT TAB 0: PAPER 6:"SOLUZIONI",
"RISPOSTE ALUNNO": FOR t=1 TO p:
LPRINT : LPRINT TAB 2,t;"":TAB
4;b$(h,t);d$(h,t);TAB 4;c$(h,t)
;e$(h,t): LPRINT : NEXT t: GO TO
8890
8863 IF p=o AND CODE w$(h)=32 TH
EN GO TO 8890
8865 IF p=o THEN INVERSE 1: LPRIN

```

```

NT TAB 0;h;"");TAB 4;a$(h);LPRI
NT : LPRINT TAB 0;"SOLUZIONI:";
INVERSE 0: LPRINT : LPRINT "L'AL
UNNO HA INDICATO UN NUMERO DI S
OLUZIONI UGUALE MA ERRAE": LPRIN
T : LPRINT TAB 0;PAPER 6;"SOLU
ZIONI";"RISPOSTE ALUNNO": FOR t=
1 TO p: LPRINT : LPRINT TAB 2;t;
");TAB 4;b$(h,t);d$(h,t);TAB 4;
c$(h,t);e$(h,t): LPRINT : NEXT t
GO TO 8890
8890 NEXT h
8892 FOR g=1 TO s: IF CODE w$(g)
<>32 THEN GO TO 8900
8894 NEXT g
8894 LPRINT : LPRINT AT 5,0;".. N
ESSUNA FRASE DA CONTROLLARE"

```

Le linee di istruzioni che consentono la procedura di controllo sopra descritta, partono da 8830 e si concludono alla linea 8852. Le risposte date dall'alunno, riguardanti il completamento del termine (soluzione) e l'accordo di questo con un altro elemento della frase, vengono memorizzate all'interno di una doppia iterazione di cui la prima riguarda le soluzioni inserite dall'insegnante, la seconda le soluzioni considerate dall'alunno. La soluzione del ragazzo, b(h,j)$, viene collocata in una memoria bidimensionale contenente come secondo indice quello relativo all'iterazione riguardante l'inserimento dei dati da parte dell'insegnante. Questo "conservare" in un'altra memoria ogni singola risposta sulla frase consente il confronto con tutte le risposte possibili previste. Solo se dal confronto emergeranno differenze nei dati, la frase contenente l'errore sarà memorizzata in $w$$.

Le istruzioni da 8855 a 8890 contengono le condizioni che consentono il confronto tra il numero delle soluzioni previste dall'insegnante e quelle indicate dall'alunno.

Queste relazioni saranno controllate solo in presenza di risposte sbagliate e sarà il confronto tra i valori, assunti dalle due variabili, "p" (numero soluzioni insegnante) ed "o" (numero soluzioni alunno), che determineranno il tipo di informazione da segnalare.

Visualizzazione dei risultati della prova

Programma in BASIC (8900÷8998)

```

8900>CLS : PRINT AT 1,0;"ALUNNO:
";i$(1 TO 24): PRINT : PRINT "C
LASSE ";i$(26 TO 35): PRINT
8910 FOR h=1 TO s
8915 LET p=VAL p$(h): LET o=VAL
o$(h)
8930 FOR j=1 TO p
8935 FOR t=1 TO o
8940 IF b$(h,j)=d$(h,t) AND c$(h
,j)=e$(h,t) THEN LET y$(h,t)=d$(
h,t): LET z$(h,t)=e$(h,t): LET d
$(h,t)=d$(h,j): LET e$(h,t)=e$(h
,j): LET d$(h,j)=y$(h,t): LET e$
(h,j)=z$(h,t): GO TO 8952
8942 NEXT t
8945 LET w$(h)=a$(h)
8950 NEXT j
8952 NEXT h
8955 IF p<o THEN INVERSE 1: PRIN
T TAB 0;h;"");TAB 4;a$(h): PRINT
: PRINT TAB 0;"SOLUZIONI:"; INV
ERSE 0: PRINT : PRINT "L'ALUNNO
HA INDICATO UN NUMERO DI SOLUZI
ONI SUPERIORE": PRINT : PRINT TA
B 0;PAPER 6;"SOLUZIONI";"RISPOS
TE ALUNNO": FOR t=1 TO o: PRINT
: PRINT TAB 2;t;");TAB 4;b$(h,t
);d$(h,t);TAB 4;c$(h,t);e$(h,t):
PRINT : NEXT t: GO TO 8990
8960 IF p>o THEN INVERSE 1: PRIN
T TAB 0;h;"");TAB 4;a$(h): PRINT
: PRINT TAB 0;"SOLUZIONI:"; INV
ERSE 0: PRINT : PRINT "L'ALUNNO

```



```

HA INDICATO UN NUMERO DI SOLUZIONI INFERIORE": PRINT : PRINT TAB 6,0; PAPER 6;"SOLUZIONI";"RISPOSTE ALUNNO": FOR t=1 TO P: PRINT : PRINT TAB 2,t;")";TAB 4;b$(h,t);d$(h,t);TAB 4;c$(h,t);e$(h,t): PRINT : NEXT t: GO TO 8990
8963 IF P=0 AND CODE w$(h)=32 THEN GO TO 8990
8965 IF P=0 THEN INVERSE 1: PRINT TAB 0,h;")";TAB 4;s$(h): PRINT : PRINT TAB 0;"SOLUZIONI"; INVERSE 0: PRINT : PRINT "L'ALUNNO HA INDICATO UN NUMERO DI SOLUZIONI UGUALE MA ERRATE": PRINT : PRINT TAB 0; PAPER 6;"SOLUZIONI";"RISPOSTE ALUNNO": FOR t=1 TO P: PRINT : PRINT TAB 2,t;")";TAB 4;b$(h,t);d$(h,t);TAB 4;c$(h,t);e$(h,t): PRINT : NEXT t: GO TO 8990
90
8990 NEXT h
8992 FOR l=1 TO 5: IF CODE w$(l) <>32 THEN GO TO 8995
8995 NEXT l
8994 PRINT : PRINT AT 5,0;"NESUNA FRASE DA CONTROLLARE "
8995 GO SUB 9500
8998 GO TO 1

```

4.3.5 vf 1/3 Ordine delle parole

I GOSUB

I GOSUB presenti in questa verifica sono già stati tutti illustrati precedentemente.

Programma in BASIC (9500÷9752)

```

9500>REM GO SUB
9501 IF INKEY$<>"" THEN GO TO 9501
9502 IF INKEY$="" THEN GO TO 9502
9503 LET x$=INKEY$: IF CODE x$<>115 AND CODE x$<>110 THEN BEEP 1,10: GO TO 9500
9504 RETURN
9505 REM GO SUB
9506 IF INKEY$<>"" THEN GO TO 9506
9507 IF INKEY$="" THEN GO TO 9507
9508 LET x$=INKEY$
9509 RETURN
9510 REM GO SUB
9511 PRINT FLASH 1;AT 21,0;s$
9512 PAUSE 0:CLS: GO TO 180
9513 REM GO SUB
9514 PRINT FLASH 1;AT 21,0;s$
9515 PAUSE 0:CLS: RETURN
9516 REM GO SUB
9517 PRINT AT 9,0;" QUESTA VERIFICA E' COSTITUITA DA "x;" FRASI " : PRINT : PRINT TAB 0;" OGNI FRASE DISPONE AL MAX. DI "92 CARATTERI."
9518 RETURN
9519 REM GO SUB
9520 PRINT AT 2,5; PAPER 6;"-----"
9521 PRINT AT 3,5; PAPER 6;"EDUCAZIONE LINGUISTICA"
9522 PRINT AT 4,5; PAPER 6;"-----"
9523 PRINT AT 5,1;"QUESTA VERIFICA E' COSTITUITA DA "s;" FRASI E SI ARTICOLA SU "TAB 1;" ESERCIZI GIÀ SVOLTI."
9524 PRINT AT 15,1;"RISPONDI ATT

```

```

ENTAMENTE ALLE "TAB 1;"DOMANDE
CHE TI VERRANNO POSTE."
9730 RETURN
9750 REM GO SUB
9751 PRINT FLASH 1;AT 20,0;"INSE
RISCI IL DATO. PREMI ENTER."
9752 RETURN

```

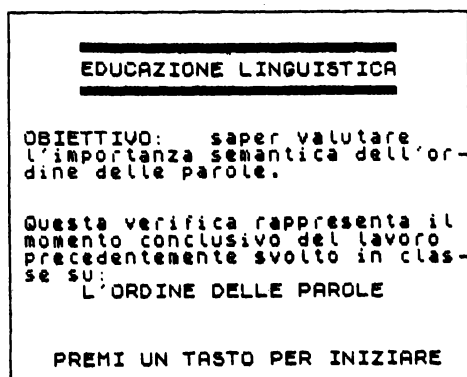
Menu

Il programma MENU si compone di tre parti:

- presentazione verifica;
- codice d'accesso;
- menù insegnanti.

Presentazione verifica

La verifica inizia con una comunicazione video che si presenta nel seguente modo:



Programma in BASIC (1÷50)

```

1>REM MENU
5 CLS : BORDER 1: PAPER 7: IN
K 0: CLS
10 PRINT AT 2,5; "-----"
5: PAPER 6; "EDUCAZIONE LINGUISTI
CA"
20 PRINT AT 4,5; "-----"
30 PRINT AT 7,1; "OBIETTIVO:
saper valutare l'importanza
semantica dell'or- dine delle pa
role."
40 PRINT AT 12,1; "Questa verif
ica rappresenta il momento conc
lusivo del lavoro precedenteme
nte svolto in clas- se su: "TAB
5: "L'ORDINE DELLE PAROLE"
50 PRINT AT 20,3; "PREMI UN TAS
TO PER INIZIARE": PAUSE 0: GO TO
60

```

Codice d'accesso

Programma in BASIC (55÷170)

```
55> PRINT AT 20,7; PAPER 6;"CO
DICE SCONOSCIUTO": PAUSE 150
60 PAUSE 100: CLS: PLOT 5,75:
DRAW PAPER 6,+245,0: DRAW PAPER
6,0,-40: DRAW PAPER 6,-245,0: D
RAW PAPER 6,0,+40
70 PLOT 5,99: DRAW PAPER 6,+24
5,0: DRAW PAPER 6,0,+40: DRAW PA
PER 6,-245,0: DRAW PAPER 6,0,-40
75 PRINT AT 7,2;"PER GLI INSEG
NANTI"----- PAPER 4;"1"
80 PRINT AT 15,2;"PER GLI ALUN
NI"----- PAPER 2; INK 7
;"2"
85 PRINT AT 1,4;"PER INIZIARE
LA VERIFICA": PRINT TAB 7;"FARSI
RICONOSCERE"
90 GO SUB 9505
100 IF CODE X$=49 THEN GO TO 15
0
110 IF CODE X$=50 THEN GO TO 80
00
120 IF CODE X$<>49 AND CODE X$<
>50 THEN BEEP 1,10: GO TO 90
150 CLS: PRINT AT 10,3; FLASH
1;"SCRIVI IL CODICE D'ACCESSO"
160 GO SUB 9505
165 IF CODE X$=45 THEN GO TO 18
0
170 IF CODE X$<>46 THEN BEEP 1,
10: GO TO 55
```

Menù insegnanti (menù principale)

Sul video comparirà il seguente menù:

EDUCAZIONE LINGUISTICA	PER CONTROLLARE LA VERIFICA
VERIFICA	LISTARE-----5
ORDINE DELLE PAROLE	CORREGGERE----2
	STAMPARE-----6
PER COSTRUIRE LA VERIFICA	PER REGISTRARE LA VERIFICA---9
INIZIALIZ ZARE-----4	
INSERIRE FRASI-----1	COMPLETATE LE OPERAZIONI PREMI----->0

Come si può notare la rappresentazione del MENU INSEGNANTI risulta simile a quella della vf 1/1; pertanto valgono le considerazioni precedentemente fatte.

Anche le fasi di lavoro saranno le stesse:

- per costruire la verifica (INIZIALIZZARE-INSERIRE);
- per controllare la verifica (LISTARE-CORREGGERE-STAMPARE);
- per registrare la verifica.

Programma in BASIC (180÷510)

```
180>CLS:PRINT AT 1,1:"PAPER 6
EDUCAZIONE":PRINT TAB 1:"PAP
ER 6; "LINGUISTICA"
190 PRINT AT 3,1; "-----"
200 PLOT 5,5
210 DRAW 0,+100: DRAW +100,0: D
RAW 0,-100: DRAW -100,0
220 PLOT 115,50
230 DRAW +135,0: DRAW 0,+40: DR
AW -135,0: DRAW 0,-40
240 PLOT 115,95
250 DRAW +135,0: DRAW 0,+75: DR
AW -135,0: DRAW 0,-75
320 PRINT AT 4,1:"VERIFICA": PR
INT:PRINT TAB 1:"PAPER 6; "ORDI
NE DELLE":PRINT TAB 1:"PAPER 6;
"PAROLE
330 PRINT AT 10,1:"PER COSTRUIR
E":PRINT AT 11,1:"LA VERIFICA"
340 PRINT AT 14,1:"INIZIALIZZ";A
T 15,1:"ZARE-----":PAPER 2; I
NK 7,"4":PRINT AT 17,1:"INSERIR
E":PRINT AT 18,1:"FRASI-----";
PAPER 2; INK 7,"1"
350 PRINT AT 12,15:"PER ":"REGI
STRARE":PRINT AT 14,15:"LA VERI
FICA---":PAPER 2; INK 7,"2"
360 PRINT AT 1,15:"PER ":"CONTR
OLLARE":PRINT AT 2,15:"LA VERIF
ICA"
380 PRINT AT 4,15:"LISTARE-----
-":PAPER 2; INK 7,"5":PRINT A
T 6,15:"CORREGGERE-----":PAPER 2
INK 7,"2":PRINT AT 8,15:"STAM
PARE-----":PAPER 2; INK 7,"6"
450 PRINT AT 16,16:"PAPER 6; "CO
MPLETATE LE":PRINT AT 19,16; PA
PER 6; "OPERAZIONI"
460 PRINT AT 20,16; PAPER 6; "PR
EMI----->0"
470 GO SUB 9505
480 IF CODE X$>57 OR CODE X$<48
THEN BEEP 1,10: GO TO 470
485 IF CODE X$=51 OR CODE X$=55
OR CODE X$=56 THEN BEEP 1,10: G
O TO 470
487 IF CODE X$=48 THEN GO TO 60
490 LET q=VAL X$
500 GO TO 1000*q
510 GO TO 1
```

Per costruire la verifica

L'insegnante che voglia costruire una nuova verifica dovrà utilizzare i seguenti sotto-programmi, in ordine di successione:

- INIZIALIZZARE;
- INSERIRE FRASI.

Inizializzare

Il programma della vf 1/3 risulta simile, nella sua struttura, a quello della vf 1/1. Esso si divide in due parti:

- a) dimensionamento della verifica;
- b) individuazione e dimensionamento delle variabili.

a. Dimensionamento della verifica

Programma in BASIC (4000÷4080)

```

4000>REM INIZIALIZZARE
4010 CLS : PRINT AT 10,3;"VUOI C
ANCELLARE LE FRASI ?";TAB 13;"(s
/n)"
4020 GO SUB 9500
4040 IF X$="s" THEN GO TO 4060
4050 GO TO 1
4060 CLS
4070 PRINT AT 10,2;"QUANTE FRASI
DEVE CONTENERE";TAB 9;"LA VERIF
ICA ?"; GO SUB 9750
4080 INPUT X

```

b. Individuazione e dimensionamento delle variabili

Poiché la verifica su l'ORDINE DELLE PAROLE si basa sui diversi significati che una frase può assumere a seconda del diverso ordine dei suoi elementi, per l'individuazione delle variabili sarà opportuno definire gli elementi che dovranno essere inseriti:

- il numero delle frasi alternative che l'insegnante dovrà costruire;
- l'elemento (o il gruppo di elementi) da spostare nel costruire le frasi;
- il giudizio di accettabilità sulle frasi alternative;
- la risposta alla domanda di verifica sulla frase alternativa ritenuta accettabile;
- il significato assunto dalla frase alternativa, confrontandola con quella di base;
- gli eventuali errori commessi dall'alunno durante la prova.

Legenda delle variabili

VARIABILE	FUNZIONE	DIMENSIONE DELLA MEMORIA
a\$(x,92)	vedi a\$ di vf1/1	vedi a\$ di vf1/1
b\$(x, 4,92)	conserverà la frase alternativa a quella di base costruita dall'insegnante.	x:= n° max. di frasi della verifica 4:=n° max. di frasi alternative 92:=n° max. di caratteri disponibili per la frase alternativa
c\$(x,4,1)	conserverà la risposta dell'insegnante relativa al giudizio di accettabilità sulla fase alternativa (s/n)	x:=n° max. di fasi della verifica 4:=n° max. di fasi alternative 1:=n° carattere disponibile (s o n)
d\$(x,60)	conserverà la domanda di verifica formulata dall'insegnante sulla frase alternativa ritenuta accettabile	x:=n° max. di frasi della verifica 60:=n° max. di caratteri disponibili per la domanda
e\$(x,60)	conserverà la risposta dell'insegnante alla domanda di verifica	x:= n° max. di frasi della verifica 60:=n° max. di caratteri disponibili per la risposta
h\$(x,4,1)	conserverà la risposta dell'insegnante sul significato assunto dalla frase alternativa accettabile confrontata con quella di base. (s o n)	x:= n° max. di frasi della verifica 4:=n° max. di frasi alternative 1:=n° caratteri disponibile (s o n)
p\$(x,1)	conserverà il numero di frasi alternative stabilito dall'insegnante	x:=n° max. di frasi della verifica 1:=n° caratteri disponibile

/segue/

VARIABILE	FUNZIONE	DIMENSIONE DELLA MEMORIA
o\$(x,60)	conserverà l'elemento (o il gruppo di elementi) da spostare nella frase, scelto dall'insegnante nel costruire le frasi alternative	x:=n° max. di frasi della verifica 60:=n° max di caratteri disponibili per l'elemento considerato
i\$(35)	vedi i\$ di vf1/1	vedi i\$ di vf1/1
y\$(x,60)	conserverà l'elemento (o il gruppo di elementi) da spostare nella frase, scelto dall'alunno	vedi o\$
g\$(x,4,1)	conserverà la risposta dell'alunno relativa al giudizio di accettabilità sulla frase alternativa.	vedi C\$
q\$(x,60)	conserverà la risposta dell'alunno alla domanda di verifica	vedi e\$
l\$(x,4,1)	conserverà la risposta dell'alunno sul significato assunto dalla frase alternativa accettabile confrontata con quella di base.	vedi h\$
z\$(x,92)	vedi Z\$ di vf1/1	vedi z\$ di vf1/1
m\$(6,35)	conserverà informazioni utili all'esecuzione delle operazioni che dovranno essere effettuate sia dall'insegnante sia dall'alunno	6=n° max. di stringhe disponibili per le informazioni 35=n° disponibile di caratteri per informazione

Programma in BASIC (4200÷4900)

```

4200 > DIM a$(x,92)
4201 DIM b$(x,4,92)
4202 DIM c$(x,4,1)
4203 DIM d$(x,60)
4204 DIM e$(x,60)
4205 DIM h$(x,4,1)
4213 DIM p$(x,1)
4214 DIM o$(x,60)
4215 DIM i$(35)
4216 DIM y$(x,60)
4217 DIM g$(x,4,1)
4218 DIM q$(x,60)
4219 DIM l$(x,4,1)
4220 DIM z$(x,92)
4290 CLS
4295 DIM m$(6,35)
4300 LET s$="PREMI UN TASTO PER
CONTINUARE"
4301 LET m$(1)="LA FRASE E' ACCET-
TABILE"
4302 LET m$(2)="LA FRASE NON E'
ACCETTABILE"
4305 LET m$(5)="LA FRASE HA LO S-
TESO SIGNIFICATO"
4890 GO SUB 9700
4900 GO TO 9550

```

Inserire frasi

- Il programma si suddivide, come per la verifica vf 1/1, in:
- determinazione del numero delle frasi da inserire;
 - inserimento frasi.

a. Determinazione del numero delle frasi da inserire

Programma in BASIC (1000÷1101)

```
1000>REM INSERIRE
1010 CLS
1090 GO SUB 9700
1091 GO SUB 9600
1100 PRINT AT 10,2;"QUANTE FRASI
    UUOI INSERIRE ?"; GO SUB 9750
1101 INPUT r
```

b. Inserimento frasi

Determinato il numero delle frasi l'insegnante potrà procedere all'inserimento di queste e di tutti gli elementi definiti nella fase di individuazione e dimensionamento delle variabili.

Le comunicazioni video saranno in successione:

```
1) Luisa ha regalato a quell'a-
mico un libro tristissimo

    QUALE ELEMENTO O GRUPPO DI ELE-
    MENTI DELLA FRASE UUOI SPOSTARE?

    INSERISCI IL DATO.  PREMI ENTER.
```

```
1) Luisa ha regalato a quell'a-
mico un libro tristissimo

    E  tristissimo

    QUANTE FRASI PUOI COSTRUIRE?
```

```

1) Luisa ha regalato a quell'a-
mico UN libro tristissimo

E tristissimo

        SCRIVI LA 1 FRASE

INERISCI IL DATO.  PREMI ENTER.

```

```

1) Luisa ha regalato a quell'a-
mico UN libro tristissimo

E tristissimo

1F Luisa ha regalato a quell'a-
mico tristissimo UN libro

        LA FRASE E' ACCETTABILE ?
        (s/n)

```

Se la frase risulterà non accettabile, la risposta verrà memorizzata e si passerà a considerare la successiva frase alternativa.

Viceversa comparirà la seguente scritta:

```

1) Luisa ha regalato a quell'a-
mico UN libro tristissimo

E tristissimo

1F Luisa ha regalato a quell'a-
mico tristissimo UN libro

A s
D QUALE TERMINE SI ACCORDA CON
  tristissimo ?

PONI LA RISPOSTA DI VERIFICA
INERISCI IL DATO.  PREMI ENTER.

```

L'accorgimento da noi adottato permette all'insegnante di poter formulare una domanda alla quale l'alunno dovrà rispondere, dimostrando di aver colto in modo consapevole il termine che si accorda con l'elemento spostato.

La risposta di verifica verrà inserita dopo la seguente scritta:

```
1) Luisa ha regalato a quell'a-  
mico Un libro tristissimo  
  
E tristissimo  
  
1F Luisa ha regalato a quell'a-  
mico tristissimo un libro  
  
A s  
  
PONI LA DOMANDA DI VERIFICA  
INSERISCI IL DATO. PREMI ENTER.  
↑
```

L'ultimo dato che l'insegnante dovrà fornire nell'esecuzione del sottoprogramma INSERIMENTO FRASI sarà relativo all'ultima richiesta che gli verrà posta:

```
1) Luisa ha regalato a quell'a-  
mico Un libro tristissimo  
  
E tristissimo  
  
1F Luisa ha regalato a quell'a-  
mico tristissimo un libro  
  
A s  
D QUALE TERMINE SI ACCORDA CON  
tristissimo ?  
R amico  
LA FRASE HA LO STESSO  
SIGNIFICATO? (s/n)
```

Completate le operazioni d'inserimento, sul video sarà visualizzata la frase con i relativi dati:

```
1) Luisa ha regalato a quell'a-  
mico Un libro tristissimo  
  
E tristissimo  
  
1F Luisa ha regalato a quell'a-  
mico tristissimo un libro  
  
A s  
D QUALE TERMINE SI ACCORDA CON  
tristissimo ?  
R amico  
S n  
PREMI UN TASTO PER CONTINUARE
```

Programma in BASIC (1103÷1999)

```

1103>FOR i=1 TO r
1105 FOR h=1 TO x
1107 IF CODE a$(h)=32 THEN GO TO
1110
1108 NEXT h
1109 CLS : PRINT AT 10,4;"VERIFI
CA COMPLETA": GO TO 9550
1110 PRINT AT 10,1;"SCRIVI LA "
h;" FRASE DA INSERIRE": PRINT FL
ASH 1;AT 21,28;"↑": GO SUB 9750
1120 INPUT a$(h)
1130 CLS
1135 INVERSE 1: PRINT TAB 0;h;"")
"TAB 4; a$(h): INVERSE 0: PRINT
1140 PRINT AT 10,1;"QUALE ELEMEN
TO 0 GRUPPO DI ELE-": TAB 0;"MEN
TI DELLA FRASE VUOI SPOSTARE?"
1143 GO SUB 9750
1145 INPUT o$(h)
1146 CLS
1147 INVERSE 1: PRINT TAB 0;h;"")
"TAB 4; a$(h): INVERSE 0: PRINT
: PRINT TAB 1;"E";TAB 4;o$(h)
1148 PRINT AT 10,2;"QUANTE FRASI
PUOI COSTRUIRE?"
1149 GO SUB 9505
1150 IF CODE x$<49 OR CODE x$>53
THEN BEEP 1,10: GO TO 1149
1151 LET p$(h)=x$
1152 LET p=VAL p$(h)
1153 CLS
1154 FOR j=1 TO p
1155 INVERSE 1: PRINT TAB 0;h;"")
"TAB 4; a$(h): INVERSE 0: PRINT
: PRINT TAB 1;"E";TAB 4;o$(h)
1156 PRINT AT 10,7;"SCRIVI LA "
j;" FRASE"
1158 PRINT FLASH 1;AT 21,28;"↑":
GO SUB 9750
1160 INPUT b$(h,j)
1161 CLS
1163 INVERSE 1: PRINT TAB 0;h;"")
"TAB 4; a$(h): INVERSE 0: PRINT
: PRINT TAB 1;"E";TAB 4;o$(h): P
RINT : PRINT TAB 0;j;"F";TAB 4;b
$(h,j)
1165 PRINT AT 18,3; a$(1): PRINT
AT 18,27;"?":TAB 13;"(s/n)"
1168 GO SUB 9500
1169 IF x$<>"s" AND x$<>"n" THEN
BEEP 1,10: GO TO 1168
1170 LET c$(h,j)=x$
1173 IF x$="n" THEN GO TO 1500
1175 CLS
1176 INVERSE 1: PRINT TAB 0;h;"")
"TAB 4; a$(h): INVERSE 0: PRINT
: PRINT TAB 1;"E";TAB 4;o$(h): P
RINT : PRINT TAB 0;j;"F";TAB 4;b
$(h,j): PRINT : PRINT TAB 1;"A";
TAB 4;c$(h,j)
1190 PRINT AT 18,3;"PONI LA DOMA
NDA DI VERIFICA";TAB 13;" "
1193 PRINT FLASH 1;AT 21,28;"↑":
GO SUB 9750
1195 INPUT d$(h)
1200 CLS
1203 INVERSE 1: PRINT TAB 0;h;"")
"TAB 4; a$(h): INVERSE 0: PRINT
: PRINT TAB 1;"E";TAB 4;o$(h): P
RINT : PRINT TAB 0;j;"F";TAB 4;b
$(h,j): PRINT : PRINT TAB 1;"A";
TAB 4;c$(h,j): PRINT : PRINT TA
B 1;"D";TAB 4;d$(h)
1205 PRINT AT 18,1;"PONI LA RISP
OSTA DI VERIFICA"
1207 PRINT FLASH 1;AT 21,28;"↑":
GO SUB 9750
1210 INPUT e$(h)
1220 CLS
1225 INVERSE 1: PRINT TAB 0;h;"")
"TAB 4; a$(h): INVERSE 0: PRINT
: PRINT TAB 1;"E";TAB 4;o$(h): P
RINT : PRINT TAB 0;j;"F";TAB 4;b
$(h,j): PRINT : PRINT TAB 1;"A";
TAB 4;c$(h,j): PRINT : PRINT TA
B 1;"D";TAB 4;d$(h): PRINT : PRI

```

```

NT TAB 1;"R";TAB 4;e$(h)
1230 PRINT AT 18,6;"LA FRASE HA
LO STESSO ";TAB 8;"SIGNIFICATO?"
TAB 22;"(s/n)"
1240 GO SUB 9500
1250 LET h$(h,j)=x$
1255 CLS
1260 INVERSE 1: PRINT TAB 0;h;"
";TAB 4;e$(h): INVERSE 0: PRINT
PRINT TAB 1;"E";TAB 4;o$(h): P
RINT: PRINT TAB 0;j;"F";TAB 4;b
$(h,j): PRINT: PRINT TAB 1;"A"
TAB 4;c$(h,j): PRINT: PRINT TA
B 1;"D";TAB 4;d$(h): PRINT: PRI
NT TAB 1;"R";TAB 4;e$(h): PRINT
PRINT TAB 1;"S";TAB 4;h$(h,j)
1270 GO SUB 9500
1500 CLS
1550 NEXT j
1900 NEXT i
1999 GO TO 180

```

Dei seguenti sottoprogrammi LISTARE-CORREGGERE-STAMPARE riportiamo solo i listati per mostrare al lettore gli aspetti peculiari della presente verifica. L'impostazione dei programmi è la stessa utilizzata per i corrispondenti programmi della vf 1/1

Listare

Programma in BASIC (5000÷5190)

```

5000>REM LISTARE
5105 CLS
5110 FOR h=1 TO x
5120 IF CODE a$(h)=32 THEN GO TO
5180
5130 LET p=VAL p$(h)
5140 FOR j=1 TO p
5143 IF c$(h,j)="s" THEN GO TO 5
153
5150 INVERSE 1: PRINT TAB 0;h;"
";TAB 4;e$(h): INVERSE 0: PRINT
PRINT TAB 1;"E";TAB 4;o$(h): P
RINT: PRINT TAB 0;j;"F";TAB 4;b
$(h,j): PRINT: PRINT TAB 1;"A"
TAB 4;c$(h,j)
5151 GO TO 5155
5155 INVERSE 1: PRINT TAB 0;h;"
";TAB 4;e$(h): INVERSE 0: PRINT
PRINT TAB 1;"E";TAB 4;o$(h): P
RINT: PRINT TAB 0;j;"F";TAB 4;b
$(h,j): PRINT: PRINT TAB 1;"A"
TAB 4;c$(h,j): PRINT: PRINT TA
B 1;"D";TAB 4;d$(h): PRINT: PRI
NT TAB 1;"R";TAB 4;e$(h): PRINT
PRINT TAB 1;"S";TAB 4;h$(h,j)
5155 GO SUB 9500
5160 CLS
5175 NEXT j
5177 CLS
5180 NEXT h
5190 GO TO 180

```

Correggere

Programma in BASIC (2000÷2400)

```

2000>REM CORREGGERE
2100 CLS: PRINT AT 10,0;"SCRIVI
IL NUMERO DELLA FRASE DA";TAB 1
0;"CORREGGERE:"; GO SUB 9750
2105 LET f=1
2110 INPUT h
2112 IF h<=x THEN GO TO 2120
2115 CLS: PRINT AT 10,2;"LA FRA
SE ";h;" NON E' INSERITA."; GO T
O 9550

```

```

2120 IF CODE a$(h)=32 THEN GO TO
2115
2125 CLS
2130 INVERSE 1: PRINT TAB 0;h;"
";TAB 4;a$(h): INVERSE 0
2135 FOR t=1 TO 4: LET o$(h)=""
LET p$(h)="" : LET b$(h,t)="" : L
ET c$(h,t)="" : LET d$(h)="" : LET
e$(h)="" : LET h$(h,t)="" : NEXT
t
2140 GO TO 1110
2400 GO TO 180

```

Stampare

Programma in BASIC (6000÷6190)

```

6000>REM STAMPARE
6105 CLS
6110 FOR h=1 TO x
6120 IF CODE a$(h)=32 THEN GO TO
6180
6130 LET p=VAL p$(h)
6135 INVERSE 1: LPRINT TAB 0;h;"
";TAB 4;a$(h): INVERSE 0: LPRIN
T : LPRINT TAB 1;"E";TAB 4;o$(h)
6140 FOR j=1 TO p
6143 IF c$(h,j)="s" THEN GO TO 6
153
6150 LPRINT : LPRINT TAB 0;j;"F"
;TAB 4;b$(h,j): LPRINT : LPRINT
TAB 1;"A";TAB 4;c$(h,j)
6151 GO TO 6155
6153 LPRINT : LPRINT TAB 0;j;"F"
;TAB 4;b$(h,j): LPRINT : LPRINT
TAB 1;"A";TAB 4;c$(h,j): LPRINT
: LPRINT : TAB 1;"D";TAB 4;d$(h):
LPRINT : LPRINT TAB 1;"R";TAB 4
;e$(h): LPRINT : LPRINT TAB 1;"S
";TAB 4;h$(h,j)
6160 LPRINT : LPRINT TAB 14;"***
*": LPRINT
6170 NEXT j
6175 LPRINT
6180 NEXT h
6190 GO TO 180

```

Riportiamo di seguito l'elenco delle frasi da noi proposto per la verifica vf 1/3

1) Luisa ha regalato a quell'a-
mico un libro tristissimo.

E tristissimo

IF Luisa tristissimo ha regala-
to a quell'amico un libro.

A n

2F Luisa ha regalato tristissi-
mo a quell'amico un libro.

A n

3F Luisa ha regalato a quell'a-
mico tristissimo un libro.

A s

D QUALE TERMINE SI ACCORDA CON
tristissimo?

R amico

S n

2) Marco ha rotto quel vaso col
fiore giallo.

E giallo

1F. Marco giallo ha rotto quel vaso col fiore.

A n

2F. Marco ha rotto giallo quel vaso col fiore.

A n

3F. Marco ha rotto quel vaso giallo col fiore.

A s

D QUALE TERMINE SI ACCORDA CON giallo ?

R vaso

S n

3) Nel bosco un cane ha ucciso un feroce cinghiale.

E feroce

1F. Nel bosco feroce un cane ha ucciso un cinghiale.

A n

2F. Nel bosco un cane feroce ha ucciso un cinghiale.

A s

D QUALE TERMINE SI ACCORDA CON feroce ?

R cane

S n

4) La maestra ha detto all'alunno di non uscire.

E non

1F. La maestra non ha detto all'alunno di uscire

A s

D QUALE VERBO SI ACCORDA CON non ? Usa l'infinito

R dire

S n

2F. La maestra ha detto non all'alunno di uscire.

A n

3F. La maestra ha detto all'alunno non di uscire.

A n

5) Prendi l'anello antico che trovi sul tavolo.

E antico

1F. Prendi antico l'anello che trovi sul tavolo.

A n

2F. Prendi l'anello che trovi antico sul tavolo.

A n

3F. Prendi l'anello che trovi sul tavolo antico

A s

D QUALE TERMINE SI ACCORDA CON antico ?

R tavolo

S n

6) Ho rielaborato con Paolo le ricerche che ho fatto.

E con Paolo

1F. Ho rielaborato le ricerche con Paolo che ho fatto.

A n

2F. Ho rielaborato le ricerche che ho fatto con Paolo.

A s

D QUALE VERBO SI ACCORDA CON con Paolo ? Usa l'infinito

R fare

S n

7) Ieri ho visto un film francese con un amico.

E francese

1F Ieri francese ho visto un film con un amico.

A n

2F Ieri ho visto francese un film con un amico.

A n

3F Ieri ho visto un film con un amico francese.

A s

D QUALE TERMINE SI ACCORDA CON francese ?

R amico

S n

8) Il re ordina l'occupazione delle regioni e punisce i sudditi ribelli.

E ribelli

1F Il re ribelli ordina l'occupazione delle regioni e punisce i sudditi.

A n

2F Il re ordina ribelli l'occupazione delle regioni e punisce i sudditi.

A n

3F Il re ordina l'occupazione delle regioni ribelli e punisce i sudditi.

A s

D QUALE TERMINE SI ACCORDA CON sudditi ?

R regioni

S n

9) Trascrivi le frasi di quel brano e sottolinea in rosso.

E in rosso

1F Trascrivi in rosso le frasi di quel brano e sottolinea.

A s

D QUALE VERBO SI ACCORDA CON in rosso ? Usa l'infinito

R trascrivere

S n

2F Trascrivi le frasi di quel in rosso brano e sottolinea.

A n

10) Quell'uomo siede davanti a quella sedia con la gamba rotta.

E con la gamba rotta

1F Quell'uomo con la gamba rotta siede davanti a quella sedia.

A s

D QUALE TERMINE SI ACCORDA CON con la gamba rotta ?

R uomo

S n

2F Quell'uomo siede davanti con la gamba rotta a quella sedia.

A n

Registrazione

Programma in BASIC (9000÷9299)

```
9000>REM REGISTRARE
9005 CLS
9010 PRINT AT 10,0;"PER REGISTRA
RE CON REG.->PREMI 1"
9015 PRINT AT 12,0;"PER REGISTRA
RE CON MIC.->PREMI 2"
9020 GO SUB 9505
9022 IF x$="1" THEN GO TO 9100
9023 IF x$="2" THEN GO TO 9200
9024 BEEP 1,10: GO TO 9010
9100 SAVE "vf1/3" LINE 1
9199 GO TO 1
9200 ERASE "m";1;"vf1/3"
9205 CLS : PLOT 20,92: DRAW +210
,0: DRAW 0,15: DRAW -210,0: DRAW
0,-15: PRINT AT 9,3;"ORA REGIST
RO LA VERIFICA."
9210 SAVE "m";1;"vf1/3" LINE 1
9299 GO TO 1
```

Eseguire

Le frasi operative del sottoprogramma ESEGUIRE risulteranno le stesse della vf 1/1:

- azzeramento delle memorie;
- memorizzazione dei dati dell'alunno;
- esecuzione della prova;
- stampa dei risultati della prova;
- visualizzazione dei risultati della prova.

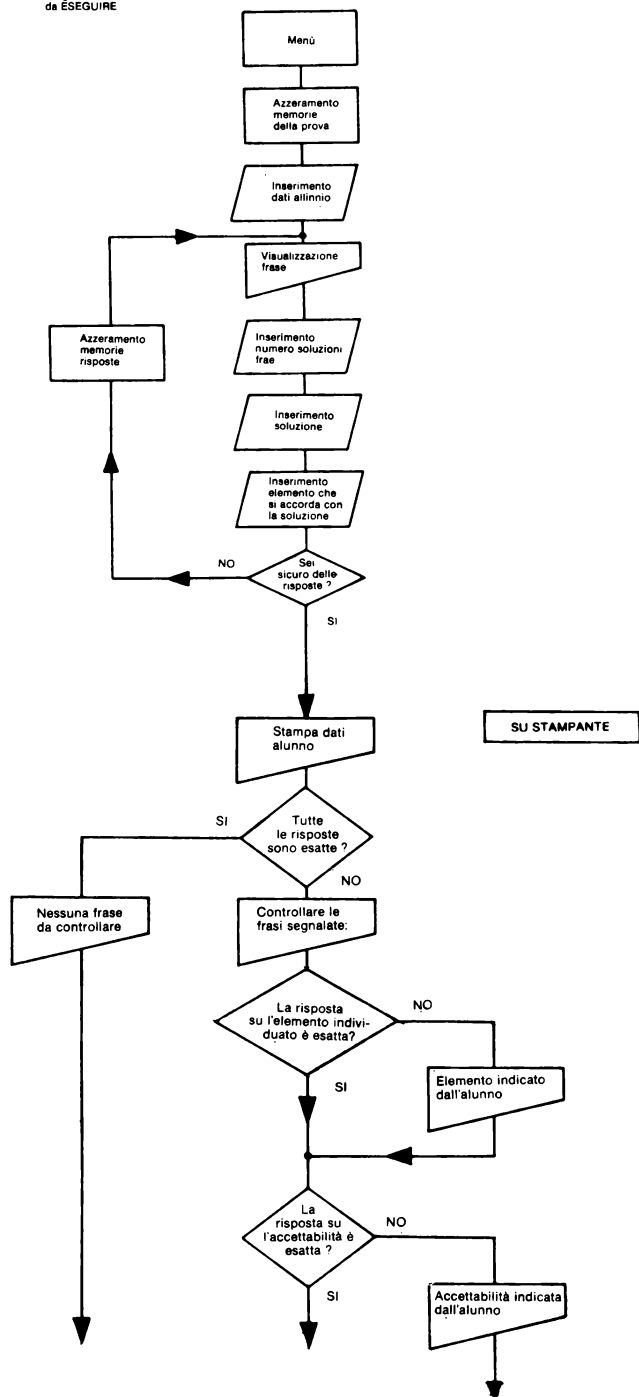
Lo schema a blocchi è dato in fig. 4.3.89.

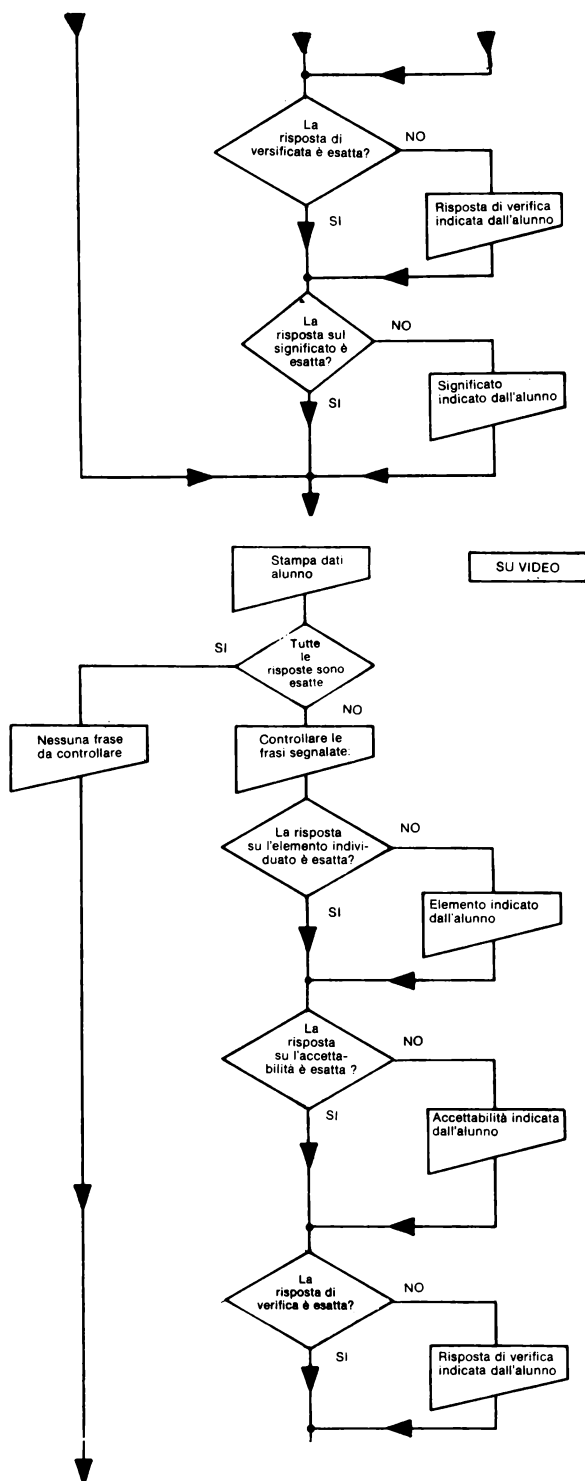
Azzeramento delle memorie

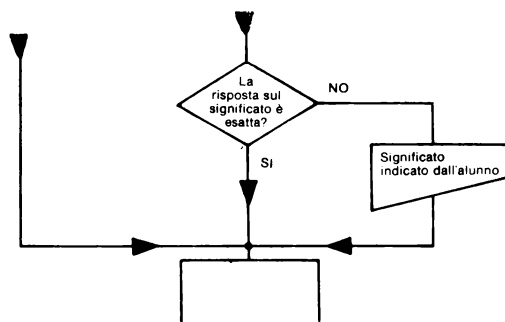
Programma in BASIC (8000÷8011).

```
8000>REM ESEGUIRE
8001 CLS
8002 FOR h=1 TO x
8003 IF CODE a$(h)=32 THEN GO TO
8005
8004 NEXT h
8005 LET s=h-1
8006 FOR h=1 TO x
8007 FOR j=1 TO 4: LET z$(h)=""
LET y$(h)="" : LET g$(h,j)="" : L
ET q$(h)="" : LET l$(h,j)="" : LET
i$="" : NEXT j
8008 NEXT h
8010 GO SUB 9710
8011 GO SUB 9500
```

Le linee di istruzione del programma in BASIC sono le stesse utilizzate per la vf 1/1; le uniche variazioni sono costituite dai diversi indirizzi di memoria definiti per la prova.







Memorizzazione dei dati dell'alunno

Programma in BASIC (8013÷8036)

Esecuzione della prova

Durante lo svolgimento della prova l'alunno sarà invitato a rispondere a numerosi quesiti.

La prima comunicazione video presenterà la frase di base e le frasi alternative costruite su di essa:

```

1) Luisa ha regalato a quell'a-
mico un libro tristissimo.

FRASI:

1) Luisa tristissimo ha regala-
to a quell'amico un libro.

2) Luisa ha regalato tristissi-
mo a quell'amico un libro.

3) Luisa ha regalato a quell'a-
mico tristissimo un libro.

PREMI UN TASTO PER CONTINUARE
  
```

Successivamente saranno riproposte, una alla volta, le frasi alternative e su ognuna l'alunno dovrà indicare:

```

1) Luisa tristissimo ha regala-
to a quell'amico un libro.

LA FRASE E' ACCETTABILE ?
(s/n)
  
```

Memorizzate le risposte, le frasi saranno nuovamente visualizzate e l'alunno, confrontandole, dovrà individuare l'elemento che, spostato, ha generato le frasi alternative:

```

1) Luisa ha regalato a quell'a-
mico un libro tristissimo.

FRASI:

1) Luisa tristissimo ha regala-
to a quell'amico un libro.
2) Luisa ha regalato tristissi-
mo a quell'amico un libro.
3) Luisa ha regalato a quell'a-
mico tristissimo un libro.

CONFRONTANDO LA FRASE BASE CON
LE FRASI SUCCESSIVE, QUAL E'
L'ELEMENTO SPOSTATO ?

INSERISCI IL DATO. PREMI ENTER.

```

Tra le frasi alternative una risulterà sicuramente accettabile: se non sarà stata individuata dall'alunno la prova, relativa alla frase base in oggetto, si interromperà memorizzando la frase e passando all'analisi di quella successiva. Viceversa, sulla frase alternativa accettabile, saranno poste altre due domande:

```

3) Luisa ha regalato a quell'a-
mico tristissimo un libro.

D QUALE TERMINE SI ACCORDA CON
tristissimo ?

INSERISCI IL DATO. PREMI ENTER.

```

```

1) Luisa ha regalato a quell'a-
mico un libro tristissimo.

FRASI:

3) Luisa ha regalato a quell'a-
mico tristissimo un libro.

LE FRASI HANNO LO STESSO
SIGNIFICATO? (s/n)

```

L'esattezza delle risposte confermerà se le valutazioni fatte sulle frasi sono state espresse in modo consapevole dall'alunno.

Programma in Basic (8040÷8251)

```

8040 FOR h=1 TO s
8050 LET p=VAL P$(h)
8060 INVERSE 1: PRINT TAB 0;h;"
";TAB 4;b$(h): INVERSE 0: PRINT
: INVERSE 1: PRINT TAB 0;"FRASI:
: INVERSE 0: PRINT
8062 FOR t=1 TO p: PRINT TAB 2;t
";":TAB 4;b$(h,t): PRINT : NEXT
t
8070 GO SUB 9600

```

```

8075 CLS
8107 FOR J=1 TO p
8110 INVERSE 1: PRINT J;"":TAB
4;b$(h,j): INVERSE 0: PRINT AT 1
0,3;"LA FRASE E' ACCETTABILE ?";
TAB 13;"(s/n)"
8115 GO SUB 9500
8117 CLS
8120 LET q$(h,j)=x$
8123 NEXT J
8124 INVERSE 1: PRINT TAB 0;h;"":
TAB 4;a$(h): INVERSE 0: PRINT
: INVERSE 1: PRINT TAB 0;"FRASI:
: INVERSE 0: PRINT
8125 FOR J=1 TO p
8127 PRINT TAB 0;J;"":TAB 4;b$(
h,j)
8128 NEXT J
8130 PRINT AT 16,1;"CONFRONTANDO
LA FRASE BASE CON";TAB 3;"LE FR
ASI SUCCESSIVE QUAL'E";TAB 5;"L
ELEMENTO SPOSTATO ?"
8131 GO SUB 9750
8132 INPUT y$(h)
8134 IF o$(h)<>y$(h) THEN GO TO
8190
8135 CLS
8137 FOR J=1 TO p
8138 IF c$(h,j)="s" THEN INVERSE
1: PRINT TAB 0;J;"":TAB 4;b$(h
,j): INVERSE 0
8139 NEXT J
8140 PRINT AT 10,1;"D";TAB 4;d$(
h)
8145 GO SUB 9750
8150 INPUT q$(h)
8155 CLS
8156 INVERSE 1: PRINT TAB 0;h;"":
TAB 4;a$(h): INVERSE 0: PRINT
: INVERSE 1: PRINT TAB 0;"FRASI:
: INVERSE 0: PRINT
8157 FOR J=1 TO p
8158 IF c$(h,j)="s" THEN PRINT T
AB 0;J;"":TAB 4;b$(h,j): GO TO
8170
8159 NEXT J
8170 PRINT AT 18,4;"LE FRASI HAN
NO LO STESSO ";TAB 8;"SIGNIFICAT
O?";TAB 22;"(s/n)"
8175 GO SUB 9500
8180 LET l$(h,j)=x$
8190 CLS
8192 PRINT AT 10,2;"SEI SICURO D
ELLE RISPOSTE ?";TAB 13;"(s/n)"
8194 GO SUB 9500
8196 IF x$="s" THEN GO TO 8250
8198 FOR J=1 TO 4: LET z$(h)="":
LET y$(h)="": LET q$(h,j)="": L
ET q$(h)="": LET l$(h,j)="": NEX
T J
8200 CLS : GO TO 8060
8250 CLS
8251 NEXT h

```

Stampa dei risultati della prova

Le informazioni che intendiamo ottenere in questa verifica sono:

- le frasi contenenti risposte errate;
- la risposta dell'alunno sulle domande di verifica;
- la risposta dell'alunno sulla diversità di significato della frase ritenuta accettabile, tra quelle proposte.

Come per le altre verifiche, il computer non considera le risposte esatte, ma segnalerà solo quelle da controllare.

Programma in BASIC (8300÷8540)

```

8300>LPRINT AT 1,0;"ALUNNO: ";i$(
1 TO 24); LPRINT : LPRINT "CLAS
SE: ";i$(26 TO 35)
8305 LET n=0
8310 FOR h=1 TO s
8320 LET p=VAL p$(h)
8330 FOR j=1 TO p
8335 IF o$(h)<>y$(h) OR c$(h,j)<
>g$(h,j) OR e$(h)<>q$(h) OR h$(h
,j)<>l$(h,j) THEN LET z$(h)=a$(h
,j) : LET n=1
8337 NEXT j
8339 NEXT h
8343 IF n=0 THEN LPRINT : LPRINT
AT 5,0;"NESSUNA FRASE DA CONT
ROLLARE "; GO TO 8500
8345 LPRINT : LPRINT AT 5,0;"CO
NTROLLARE LE FRASI SEGNALATE "
LPRINT
8346 FOR h=1 TO s
8347 IF CODE z$(h)=32 THEN GO TO
8540
8348 LPRINT : INVERSE 1: LPRINT
TAB 0,h,j);TAB 4,z$(h); INVERSE
0: LPRINT : LPRINT TAB 0;"L'ALU
NNO HA INDICATO:" LPRINT
8350 IF o$(h)=y$(h) THEN GO TO 8
355
8352 LPRINT TAB 1;"E";TAB 4;y$(h
); GO TO 8540
8355 LET p=VAL p$(h)
8360 FOR j=1 TO p
8365 IF c$(h,j)="n" AND c$(h,j)<
>g$(h,j) THEN LPRINT : LPRINT TA
B 0,j);"F";TAB 4,b$(h,j); LPRINT
TAB 1;"A";TAB 4,g$(h,j); GO TO 8
520
8367 IF c$(h,j)="n" AND c$(h,j)=
g$(h,j) THEN GO TO 8520
8370 IF c$(h,j)="s" AND c$(h,j)=
g$(h,j) AND e$(h)=q$(h) AND h$(h
,j)=l$(h,j) THEN GO TO 8520
8374 LPRINT : LPRINT TAB 0,j);"F"
;TAB 4;b$(h,j)
8376 IF c$(h,j)="s" AND c$(h,j)=
g$(h,j) THEN GO TO 8379
8378 LPRINT : LPRINT TAB 1;"A";T
AB 4;g$(h,j)
8379 IF c$(h,j)="s" AND e$(h)=q$(
h) THEN GO TO 8405
8400 LPRINT : LPRINT TAB 1;"R";T
AB 4;q$(h)
8405 IF c$(h,j)="s" AND h$(h,j)=
l$(h,j) THEN GO TO 8520
8410 LPRINT : LPRINT TAB 1;"S";T
AB 4;l$(h,j)
8520 LPRINT : LPRINT TAB 14;"***
*"
LPRINT
8530 NEXT j
8540 NEXT h

```

Visualizzazione dei risultati della prova

Programma in BASIC (8600÷8991)

```

8600>PRINT AT 1,0;"ALUNNO: ";i$(
1 TO 24); PRINT : PRINT "CLASSE:
";i$(26 TO 35)
8605 LET n=0
8610 FOR h=1 TO s
8620 LET p=VAL p$(h)
8630 FOR j=1 TO p
8635 IF o$(h)<>y$(h) OR c$(h,j)<
>g$(h,j) OR e$(h)<>q$(h) OR h$(h
,j)<>l$(h,j) THEN LET z$(h)=a$(h
,j) : LET n=1
8637 NEXT j
8639 NEXT h
8643 IF n=0 THEN PRINT AT 5,0;"
NESSUNA FRASE DA CONTROLLARE "
: GO TO 8990
8645 PRINT AT 5,0;" CONTROLLARE

```

```

LE FRASI SEGNALATE ": PRINT
8646 FOR h=1 TO 5
8647 IF CODE Z$(h)=32 THEN GO TO
8950
8648 PRINT : INVERSE 1: PRINT TA
B 0;h;":TAB 4;Z$(h): INVERSE 0
: PRINT : PRINT TAB 0;"L'ALUNNO
HA INDICATO:": PRINT
8650 IF O$(h)=y$(h) THEN GO TO 8
655
8652 PRINT TAB 1;"E";TAB 4;y$(h)
: GO TO 8950
8655 LET p=VAL p$(h)
8660 FOR j=1 TO p
8665 IF C$(h,j)="n" AND C$(h,j)<
>g$(h,j) THEN PRINT : PRINT TAB
0;j;"F";TAB 4;b$(h,j): PRINT TAB
1;"A";TAB 4;g$(h,j): GO TO 8800
8667 IF C$(h,j)="n" AND C$(h,j)=
g$(h,j) THEN GO TO 8800
8670 IF C$(h,j)="s" AND C$(h,j)=
g$(h,j) AND e$(h)=q$(h) AND h$(h
,j)=(s$(h,j) THEN GO TO 8800
8674 PRINT : PRINT TAB 0;j;"F";T
AB 4;b$(h,j)
8676 IF C$(h,j)="s" AND C$(h,j)=
g$(h,j) THEN GO TO 8679
8678 PRINT : PRINT TAB 1;"A";TAB
4;g$(h,j)
8679 IF C$(h,j)="s" AND e$(h)=q$(
h) THEN GO TO 8705
8700 PRINT : PRINT TAB 1;"R";TAB
4;q$(h)
8705 IF C$(h,j)="s" AND h$(h,j)=
l$(h,j) THEN GO TO 8800
8710 PRINT : PRINT TAB 1;"S";TAB
4;l$(h,j)
8800 PRINT : PRINT TAB 14;"****"
8801 NEXT j
8950 NEXT h
8990 GO SUB 9600
8991 GO TO 1

```

4.3.6 vs 1 - Verifica sommativa

Questa verifica, che rappresenta il momento di sintesi delle precedenti verifiche formative, ha lo scopo di fornire esclusivamente dei dati sul livello di apprendimento a cui è giunto l'alunno alla fine dell'intero percorso didattico. Pertanto non sarà più indispensabile specificare per ogni frase i motivi che rendono necessario un controllo (su cui strutturare il lavoro di recupero) ma offrire solo l'elenco di quelle sulle quali l'alunno non è stato in grado di formulare una corretta risposta.

La verifica sommativa proprio perché deve inglobare i vari aspetti attraverso cui è stato studiato il problema dell'ACCETTABILITÀ/NON ACCETTABILITÀ delle frasi, risulterà costituita dall'insieme delle caratteristiche che hanno permesso la costruzione delle singole verifiche formative.

I GOSUB

Programma in BASIC (9500÷9860)

```

9500>REM GO SUB
9501 IF INKEY$<>"" THEN GO TO 95
01
9502 IF INKEY$="" THEN GO TO 950
2
9503 LET x$=INKEY$: IF CODE x$<>
115 AND CODE x$<>110 THEN BEEP 1
: GO TO 9500
9504 RETURN
9505 REM GO SUB
9506 IF INKEY$<>"" THEN GO TO 95
06

```

```

9507 IF INKEY$="" THEN GO TO 950
9508 LET X$=INKEY$
9509 RETURN
9550 REM GO SUB
9551 PRINT FLASH 1;AT 21,0;S$
9552 PAUSE 0;CLS:GO TO 180
9600 REM GO SUB
9601 PRINT FLASH 1;AT 21,0;S$
9602 PAUSE 0;CLS:RETURN
9650 REM GO SUB
9651 PRINT AT 10,12;"PERCHE'",TAB
9652 PRINT AT 11,30;"?",P
9653 PRINT:PRINT:TAB 1;"SE DISORDIN
9654 ATA----->PREMI 1:PRINT:PR
9655 INT:TAB 1;"SE PER DISACCORDO",T
9656 AB 1;"GRAMMATICALE----->PRE
9657 MI 2:PRINT:PRINT:TAB 1;"SE
9658 PER DISACCORDO",TAB 1;"DI SIGNIF
9659 ICATO----->PREMI 3"
9660 RETURN
9700 REM GO SUB
9701 PRINT AT 9,0;" QUESTA VERI
9702 FICA E' COSTITUITA DA X",FR
9703 ASI:PRINT:PRINT TAB 0;" OGN
9704 I FRASE DISPONE AL MAX. DI 92
9705 CARATTERI."
9706 RETURN
9710 REM GO SUB
9711 PRINT AT 2,5;PAPER 6;"-----
9712 5;PAPER 6;"EDUCAZIONE LINGUISTI
9713 CA"
9714 PRINT AT 4,5;PAPER 6;"-----
9715 PRINT AT 8,1;"QUESTA VERIF
9716 ICA E' COSTITUITA",TAB 1;"DA",S
9717 " FRASI E SI ARTICOLA SU",TAB
9718 1;"ESERCIZI GIA' SVOLTI."
9719 PRINT AT 15,1;"RISPONDI ATT
9720 ENTAMENTE ALLE",TAB 1;"DOMANDE
9721 CHE TI VERRANNO POSTE."
9722 RETURN
9730 REM GO SUB
9731 PRINT FLASH 1;AT 20,0;"INSE
9732 RISCI IL DATO. PREMI ENTER."
9733 RETURN
9800 REM GO SUB
9801 INVERSE 1:PRINT TAB 0;h;"")
9802 TAB 4;a$(h):INVERSE 0:PRINT
9803 PRINT:INVERSE 1:PRINT TA
9804 B 0;"SOLUZIONI":INVERSE 0:PRI
9805 NT
9806 FOR t=1 TO p:PRINT TAB 2;t
9807 ";":TAB 4;b$(h,t);TAB 4;c$(h,t)
9808 :PRINT:NEXT t
9809 RETURN
9850 REM GO SUB
9851 INVERSE 1:PRINT TAB 0;h;"")
9852 TAB 4;a$(h):PRINT:PRINT TAB
9853 0;"SOLUZIONI":INVERSE 0:PRI
9854 NT
9855 FOR t=1 TO v:PRINT TAB 2;t
9856 ";":TAB 4;d$(h,t);TAB 4;e$(h,t)
9857 :PRINT:NEXT t
9858 RETURN

```

Menù

Il programma MENU si compone di tre parti:

- presentazione verifica;
- codice d'accesso;
- menù insegnanti.

Presentazione verifica

La verifica inizia con la seguente comunicazione video:

EDUCAZIONE LINGUISTICA

QUESTA VERIFICA SOMMATIVA SI
ARTICOLA SUI CONTENUTI DELLE
PRECEDENTI SCHEDE.

E' NECESSARIO AVER SVOLTO LE
VERIFICHE:

vf1/1 - vf1/2 - vf1/3.

PREMI UN TASTO PER INIZIARE

e successivamente:

VERIFICA SOMMATIVA

OBIETTIVO: saper valutare
quando una frase e'

ACCETTABILE - NON ACCETTABILE

PREMI UN TASTO PER INIZIARE

Programma in BASIC (1÷52)

```
1>REM MENU
5 CLS : BORDER 1: PAPER 7: IN
K 0: CLS
10 PRINT AT 2,5: PAPER 6: "
-----
5: PAPER 6;"EDUCAZIONE LINGUISTI
CA"
20 PRINT AT 4,5: PAPER 6: "
-----
30 PRINT AT 7,1:"QUESTA VERIFI
CA SOMMATIVA SI": PRINT : PRINT
TAB 1;"ARTICOLA SUI CONTENUTI DE
LLE": PRINT : PRINT TAB 1;"PRECE
DENTI SCHEDE."
40 PRINT : PRINT TAB 1;"E' NEC
ESSARIO AVER SVOLTO LE": PRINT :
PRINT TAB 1;"VERIFICHE": PRINT
: PRINT TAB 4;"vf1/1 - vf1/2 -
vf1/3"
50 PRINT AT 20,3:"PREMI UN TAS
TO PER INIZIARE": PAUSE 0
52 CLS : PRINT AT 2,7: PAPER 0
: INK 7;"VERIFICA SOMMATIVA": PR
INT AT 6,1;"OBIETTIVO: saper val
utare": PRINT : PRINT TAB 1;"qua
ndo una frase e'": PRINT AT 11,
2: PAPER 6;"ACCETTABILE - NON AC
CETTABILE": PRINT : PRINT AT 20,
3:"PREMI UN TASTO PER INIZIARE":
PAUSE 0: GO TO 60
```


Codice d'accesso

Programma in BASIC (55÷170)

```
55>PRINT AT 20,7;"PAPER 6:"COD
ICE SCONOSCIUTO"; PAUSE 150
60 PAUSE 100:CLS: PLOT 5,75:
DRAW PAPER 6,+245,0: DRAW PAPER
6,0,-40: DRAW PAPER 6,-245,0: D
RAW PAPER 6,0,+40
70 PLOT 5,99: DRAW PAPER 6,+24
5,0: DRAW PAPER 6,0,+40: DRAW PA
PER 6,-245,0: DRAW PAPER 6,0,-40
75 PRINT AT 7,2;"PER GLI INSEG
NANTI - - - - - PAPER 4,1"
80 PRINT AT 15,2;"PER GLI ALUN
NI - - - - -"; PAPER 2; INK 7
;"2"
85 PRINT AT 1,4;"PER INIZIARE
LA VERIFICA": PRINT TAB 7;"FARSI
RICONOSCERE"
90 GO SUB 9505
100 IF CODE X$=49 THEN GO TO 15
0
110 IF CODE X$=50 THEN GO TO 80
00
120 IF CODE X$<>49 AND CODE X$<
>50 THEN BEEP 1,10: GO TO 90
150 CLS: PRINT AT 10,3; FLASH
1;"SCRIVI IL CODICE D'ACCESSO"
160 GO SUB 9505
165 IF CODE X$=46 THEN GO TO 18
0
170 IF CODE X$<>46 THEN BEEP 1,
10: GO TO 55
```

Menù insegnanti (menù principale)

Sul video comparirà il seguente menù:

EDUCAZIONE LINGUISTICA	PER CONTROLLARE LA VERIFICA
VERIFICA	LISTARE-----5
ACCETTABILITÀ DELLE FRASI	CORREGGERE----2
	STAMPARE-----6
PER COSTRUIRE LA VERIFICA	PER REGISTRARE LA VERIFICA---9
INIZIALIZ ZARE-----4	
INSERIRE FRASI-----1	COMPLETATE LE OPERAZIONI PREMI----->0

Come si può notare la rappresentazione del MENU INSEGNANTI risulta simile a quella della vf 1/1; pertanto valgono le considerazioni precedentemente fatte.

Anche le fasi di lavoro saranno le medesime:

- per costruire la verifica (INIZIALIZZARE-INSERIRE);
- per controllare la verifica (LISTARE-CORREGGERE-STAMPARE);
- per registrare la verifica.

Programma in BASIC (180÷510)

```

180>CLS : PRINT AT 1,1; PAPER 6
"EDUCAZIONE": PRINT TAB 1; PAPER 6; "LINGUISTICA"
190 PRINT AT 3,1; "-----"
200 PLOT 5,5
210 DRAW 0,+100; DRAW +100,0; D
RAU 0,-100; DRAW -100,0
220 PLOT 115,50
230 DRAW +135,0; DRAW 0,+40; DR
AU -135,0; DRAW 0,-40
240 PLOT 115,95
250 DRAW +135,0; DRAW 0,+75; DR
AU -135,0; DRAW 0,-75
320 PRINT AT 4,1; "VERIFICA": PR
INT : PRINT TAB 1; PAPER 6; "ACCE
TTABILITA": PRINT TAB 1; PAPER 6
"DELLE FRASI": PLOT 100,120;
PLOT 100,128
330 PRINT AT 10,1; "PER COSTRUIR
E": PRINT AT 11,1; "LA VERIFICA"
340 PRINT AT 14,1; "INIZIALIZZAR
E": PAPER 2; INK 7; "1"
350 PRINT AT 12,15; "PER": "REGI
STRARE": PRINT AT 14,15; "LA VERI
FICA": PAPER 2; INK 7; "2"
360 PRINT AT 1,15; "PER": "CONTR
OLLARE": PRINT AT 2,15; "L'UNITA
"
370 PRINT AT 1,15; "PER": "CONTR
OLLARE": PRINT AT 2,15; "LA VERIF
ICA"
380 PRINT AT 4,15; "LISTARE": PAPER 2; INK 7; "5": PRINT A
T 6,15; "CORREGGERE": PAPER 2; INK 7; "2": PRINT AT 8,15; "STAM
PARE": PAPER 2; INK 7; "6"
450 PRINT AT 18,15; PAPER 6; "CO
MPLETATE LE": PRINT AT 19,15; PAPER 6; "OPERAZIONI"
460 PRINT AT 20,15; PAPER 6; "PR
EMI":>0
470 GO SUB 9505
480 IF CODE X$>57 OR CODE X$<48
THEN BEEP 1,10: GO TO 470
485 IF CODE X$=51 OR CODE X$=55
OR CODE X$=56 THEN BEEP 1,10: G
O TO 470
487 IF CODE X$=48 THEN GO TO 60
490 LET q=VAL X$
500 GO TO 1000*q
510 GO TO 1

```

Per costruire la verifica

L'insegnante che voglia costruire una nuova verifica dovrà utilizzare i seguenti sotto-programmi, in ordine di successione:

- INIZIALIZZARE;
- INSERIRE FRASI.

Inizializzare

Il programma della vs 1 risulta simile, nella sua struttura, a quello già illustrato nella vf 1/1. Esso si divide in due parti:

- a) dimensionamento della verifica;
- b) individuazione e dimensionamento delle variabili.

a. Dimensionamento della verifica

Programma in BASIC (4000÷4080)

```

4000>REM INIZIALIZZARE
4010 CLS : PRINT AT 10,3;"VUOI C
ANCELLARE LE FRASI ?";TAB 13;"(S
/n)"
4020 GO SUB 9500
4040 IF x$="s" THEN GO TO 4060
4050 GO TO 180
4060 CLS
4070 PRINT AT 10,2;"QUANTE FRASI
DEVE CONTENERE";TAB 8;"LA VERIF
ICA ?": GO SUB 9750
4080 INPUT x

```

b. Individuazione e dimensionamento delle variabili

```

4090>CLS
4200 DIM a$(x,92)
4201 DIM b$(x,4,14)
4202 DIM c$(x,4,14)
4203 DIM d$(x,4,14)
4204 DIM e$(x,4,14)
4205 DIM y$(x,4,14)
4206 DIM z$(x,4,14)
4208 DIM f$(x,10,1)
4209 DIM m$(6,35)
4210 DIM i$(36)
4211 DIM j$(x,4,92)
4212 DIM u$(x,4,1)
4213 DIM t$(x,60)
4214 DIM r$(x,60)
4215 DIM h$(x,4,1)
4216 DIM p$(x,1)
4217 DIM o$(x,60)
4218 DIM n$(x,60)
4219 DIM g$(x,4,1)
4220 DIM q$(x,60)
4221 DIM l$(x,4,1)
4222 DIM w$(x,92)
4301 LET m$(1)="LA FRASE E' ACCE
TTABILE"
4302 LET m$(2)="LA FRASE NON E'
ACCETTABILE"
4303 LET m$(3)="PERCHE' DISORDIN
ATA"
4304 LET m$(4)="PER DISACCORDO G
RAMMATICALE"
4305 LET m$(5)="PER DISACCORDO D
I SIGNIFICATO"
4306 LET m$(6)="LA FRASE HA LO S
TESSO SIGNIFICATO"
4530 LET s$=" PREMI UN TASTO PE
R CONTINUARE "
4890 GO SUB 9700
4900 GO TO 9550

```

Legenda delle variabili

Poiché la verifica sommativa conterà l'insieme dei programmi delle verifiche formative, riportiamo la lista delle variabili utilizzate per la vs 1 rimandando la loro spiegazione ai commenti già fatti nelle precedenti tabelle.

Variabili specifiche della vs1	w\$(x,92) corrisponde a z\$ di vf1/1 f\$(x,1,1) conserverà il tipo di giudizio espresso dall'insegnante sulla frase (ACCETTABILITÀ/REGOLE DI ACCORDO/ORDINE DELLE PAROLE)
Variabili comuni alle 4 verifiche: vf1/1; vf1/2; vf1/3; vs1	a\$(x,92) vedi vf1/1 i\$(36) vedi vf 1/1 m\$ (6,35) vedi vf 1/1

Variabili utilizzate per i GIUDIZI DI NON ACCETTABILITÀ	$f\$ (x,5,1)$ corrisponde a $b\$$ di $vf1/1$ $f\$ (x,2,1)$ corrisponde a $y\$$ di $vf1/1$ $f\$ (x,7,1)$ conserverà il giudizio espresso dall'alunno sull'accettabilità $f\$ (x,6,1)$ corrisponde a $c\$$ di $vf1/1$ $f\$ (x,3,1)$ conserverà il numero di soluzioni indicate dall'insegnante vedi $p\$$ di $vf1/2$ $f\$ (x,4,1)$ conserverà il numero delle soluzioni indicate dall'alunno vedi $o\$$ di $vf1/2$
Variabili utilizzate per le REGOLE DI ACCORDO	$b\$ (x,4,14)$ $c\$ (x,4,14)$ $d\$ (x,4,14)$ $e\$ (x,4,14)$ vedi le corrispondenti variabili dalla $vf1/2$ $y\$ (x,4,14)$ $z\$ (x,4,14)$
Variabili utilizzate per l'ORDINE DELLE PAROLE	$j\$ (x,4,92)$ corrisponde a $b\$$ di $vf1/3$ $u\$ (x,4,1)$ corrisponde a $c\$$ di $vf1/3$ $t\$ (x,60)$ corrisponde a $d\$$ di $vf1/3$ $r\$ (x,60)$ corrisponde a $e\$$ di $vf1/3$ $h\$ (x,4,1)$ corrisponde a $h\$$ di $vf1/3$ $p\$ (x,1)$ corrisponde a $p\$$ di $vf1/3$ $o\$ (x,60)$ corrisponde a $o\$$ di $vf1/3$ $n\$ (x,60)$ corrisponde a $y\$$ di $vf1/3$ $g\$ (x,4,1)$ corrisponde a $g\$$ di $vf1/3$ $q\$ (x,60)$ corrisponde a $q\$$ di $vf1/3$ $l\$ (x,4,1)$ corrisponde a $l\$$ di $vf1/3$

Inserire frasi

Il programma si suddivide, come per la $vf1/1$, in:
a) determinazione del numero delle frasi da inserire;
b) inserimento frasi.

a. Determinazione del numero delle frasi da inserire

Programma in BASIC (1000÷1101)

```

1000>REM INSERIRE
1010 CLS
1090 GO SUB 9700
1091 GO SUB 9600
1100 PRINT AT 10,2;"QUANTE FRASI
VUOI INSERIRE?"; GO SUB 9750
1101 INPUT r

```

b. Inserimento frasi

Determinato il numero delle frasi l'insegnante potrà procedere all'inserimento di queste e di tutti gli elementi definiti nella fase di individuazione e dimensionamento delle variabili.

L'accorgimento che consente di classificare il tipo di frase corrispondente ai tre giudizi possibili (ACCETTABILITÀ/REGOLE DI ACCORDO/ORDINE DELLE PAROLE) da noi adottato è stato sintetizzato nella seguente comunicazione video:

```

1) Al cadere delle foglie inco-
mincia la primavera

        QUALE GIUDIZIO VA ESPRESSO ?

ACCETTABILITA' -----1

REGOLE DI
ACCORDO-----2

ORDINE DELLE
PAROLE-----3

```

L'insegnante una volta espresso il giudizio dovrà compiere le operazioni specifiche delle verifiche formative a cui le frasi fanno riferimento. Anche il programma riflette questa suddivisione, è cioè strutturato in tre parti ognuna delle quali contiene le stesse istruzioni utilizzate per l'INSERIMENTO FRASI delle corrispondenti verifiche formative. Questo modo di operare semplifica le procedure di altri sottoprogrammi quali: LISTARE-STAMPARE-ESEGUIRE.

Programma in BASIC (1103÷1999).

```

1103>FOR i=1 TO r
1105 FOR h=1 TO x
1107 IF CODE a$(h)=32 THEN GO TO
1110
1108 NEXT h
1109 CLS : PRINT AT 10,6;"VERIFI
CA COMPLETA": GO TO 9550
1110 PRINT AT 10,1;"SCRIVI LA "
h;" FRASE DA INSERIRE": PRINT FL
ASH 1;AT 21,28;"f": GO SUB 9750
1120 INPUT a$(h)
1130 CLS
1135 INVERSE 1: PRINT TAB 0;h;")
";TAB 4;a$(h): INVERSE 0
1140 PRINT AT 7,2: PAPER 6;"QUAL
E GIUDIZIO VA ESPRESSO ?"
1170 PRINT AT 10,1;"ACCETTABILIT
A'-----"; PAPER 6;"1"
1180 PRINT AT 13,1;"REGOLE DI";T
AB 1;"ACCORDO-----";
"; PAPER 6;"2"
1190 PRINT AT 17,1;"ORDINE DELLE
";TAB 1;"PAROLE-----";
"; PAPER 6;"3"
1195 GO SUB 9505
1197 IF CODE x$(49 OR CODE x$>51
THEN BEEP 1,10: GO TO 1195
1198 LET f$(h,1)=x$
1200 IF CODE f$(h,1)=49 THEN GO
SUB 1300
1210 IF CODE f$(h,1)=50 THEN GO
SUB 1500
1220 IF CODE f$(h,1)=51 THEN GO
SUB 1800
1225 GO SUB 9500
1230 NEXT i
1240 GO TO 180
1300 REM GIUDIZI ACCETTABILITA'
1301 CLS
1302 INVERSE 1: PRINT TAB 0;h;")
";TAB 4;a$(h): INVERSE 0
1305 PRINT AT 10,3;#$(1): PRINT
AT 10,27;"?";TAB 13;"(s/n)"
1310 GO SUB 9500
1315 LET f$(h,5)=x$
1320 CLS
1325 INVERSE 1: PRINT TAB 0;h;")
";TAB 4;a$(h): INVERSE 0

```

```

1330 IF f$(h,5)="s" THEN PRINT A
T 4,5;m$(1): GO TO 1225
1335 GO SUB 9550
1340 GO SUB 9505
1350 IF CODE x$<49 OR CODE x$>51
THEN BEEP 1,10: GO TO 1340
1360 LET f$(h,6)=x$
1370 CLS
1380 INVERSE 1: PRINT TAB 0;h;")
"TAB 4,a$(h): INVERSE 0
1390 IF f$(h,6)="1" THEN PRINT A
T 4,2;m$(2);TAB 6;m$(3): GO TO 1
225
1400 IF f$(h,6)="2" THEN PRINT A
T 4,2;m$(2);TAB 2;m$(4): GO TO 1
225
1410 IF f$(h,6)="3" THEN PRINT A
T 4,2;m$(2);TAB 1;m$(5)
1420 GO TO 1225
1500 REM REGOLE DI ACCORDO
1505 CLS
1510 INVERSE 1: PRINT TAB 0;h;")
"TAB 4,a$(h): INVERSE 0
1520 PRINT AT 10,2;"QUANTE SOLUZ
IONI HA LA FRASE?"
1530 GO SUB 9505
1540 IF CODE x$<48 OR CODE x$>52
THEN BEEP 1,10: GO TO 1520
1550 LET f$(h,3)=x$
1560 LET p=VAL f$(h,3)
1570 CLS
1580 FOR j=1 TO p
1590 INVERSE 1: PRINT AT 0,0;h;"
"TAB 4,a$(h): INVERSE 0: PRINT
AT 18,5;"SCRIVI LA ",j;" SOLUZI
ONE"
1600 GO SUB 9750
1610 INPUT b$(h,j)
1620 CLS
1630 GO SUB 9800
1640 PRINT AT 17,0;" SCRIVI L'EL
EMENTO DELLA FRASE",TAB 5;"CHE
SI ACCORDA CON LA ",TAB 10;j;" S
OLUZIONE"
1650 GO SUB 9750
1660 INPUT c$(h,j)
1670 CLS
1680 GO SUB 9800
1690 NEXT j
1710 GO TO 1225
1800 REM ORDINE PAROLE
1803 CLS
1804 INVERSE 1: PRINT TAB 0;h;")
"TAB 4,a$(h): INVERSE 0
1805 PRINT AT 10,1;"QUALE ELEMEN
TO O GRUPPO DI ELE-",TAB 0;"MENT
I DELLA FRASE VUOI SPOSTARE?"
1810 PRINT FLASH 1;AT 21,28;"↑":
GO SUB 9750
1815 INPUT o$(h)
1817 CLS
1820 INVERSE 1: PRINT TAB 0;h;")
"TAB 4,a$(h): INVERSE 0: PRINT
PRINT TAB 0;"E";TAB 4;o$(h)
1823 PRINT AT 10,2;"QUANTE FRASI
PUOI COSTRUIRE?"
1824 GO SUB 9505
1826 IF CODE x$<49 OR CODE x$>53
THEN BEEP 1,10: GO TO 1824
1830 LET p$(h)=x$
1831 LET p=VAL p$(h)
1833 CLS
1834 FOR j=1 TO p
1835 INVERSE 1: PRINT TAB 0;h;")
"TAB 4,a$(h): INVERSE 0: PRINT
PRINT TAB 1;"E";TAB 4;o$(h)
1836 PRINT AT 10,7;"SCRIVI LA ",
j;" FRASE"
1838 PRINT FLASH 1;AT 21,28;"↑":
GO SUB 9750
1840 INPUT j$(h,j)
1841 CLS
1843 INVERSE 1: PRINT TAB 0;h;")
"TAB 4,a$(h): INVERSE 0: PRINT
PRINT TAB 1;"E";TAB 4;o$(h): P
RINT: PRINT TAB 0;j;"F";TAB 4;j
$(h,j)
1845 PRINT AT 18,3;m$(1): PRINT
AT 18,27;"?";TAB 13;"(s/n)"
1848 GO SUB 9500

```

```

1849 IF x$<>"s" AND x$<>"n" THEN
  BEEP 1,10: GO TO 1848
1850 LET u$(h,j)=x$
1853 IF x$="n" THEN GO TO 1970
1855 CLS
1856 INVERSE 1: PRINT TAB 0,h:")
  TAB 4,3$(h): INVERSE 0: PRINT
  PRINT TAB 1,"E":TAB 4,0$(h): P
  RINT: PRINT TAB 0,j,"F":TAB 4,j
  $(h,j): PRINT: PRINT TAB 1,"A":
  TAB 4,u$(h,j): PRINT: PRINT TAB
  1,"D":TAB 4,t$(h): PRINT: PRIN
  T TAB 1,"R":TAB 4,r$(h): PRINT:
  1900 PRINT AT 18,6:"LA FRASE HA
  LO STESSO";TAB 8:"SIGNIFICATO?"
  TAB 22,"(s/n)"
1905 GO SUB 9500
1910 LET h$(h,j)=x$
1915 CLS
1920 INVERSE 1: PRINT TAB 0,h:")
  TAB 4,3$(h): INVERSE 0: PRINT
  PRINT TAB 1,"E":TAB 4,0$(h): P
  RINT: PRINT TAB 0,j,"F":TAB 4,j
  $(h,j): PRINT: PRINT TAB 1,"A":
  TAB 4,u$(h,j): PRINT: PRINT TAB
  1,"D":TAB 4,t$(h): PRINT: PRIN
  T TAB 1,"R":TAB 4,r$(h): PRINT:
  PRINT TAB 1,"S":TAB 4,h$(h,j)
1930 GO SUB 9500
1970 CLS
1975 NEXT J
1990 GO TO 1230

```

Listare

Programma in BASIC (5000÷5190)

```

5000>REM LISTARE
5010 CLS
5020 FOR h=1 TO x
5030 IF CODE a$(h)=32 THEN GO TO
5050
5045 IF f$(h,1)="1" THEN GO TO 5
050
5047 IF f$(h,1)="2" THEN GO TO 5
250
5049 IF f$(h,1)="3" THEN GO TO 5
430
5050 INVERSE 1: PRINT TAB 0,h:")
  TAB 4,3$(h): INVERSE 0: PRINT
5055 IF f$(h,5)="s" THEN PRINT T
AB 4,m$(1): GO SUB 9500: GO TO 5
300
5060 IF f$(h,6)="1" THEN PRINT T
AB 2;m$(2);TAB 6;m$(3)
5070 IF f$(h,6)="2" THEN PRINT T
AB 2;m$(2);TAB 2;m$(4)
5080 IF f$(h,6)="3" THEN PRINT T
AB 2;m$(2);TAB 1;m$(5)

```

```

5090 GO SUB 9600: GO TO 5800
5250 INVERSE 1: PRINT TAB 0;h;"")
"TAB 4;a$(h): INVERSE 0: PRINT
5255 PRINT : PRINT TAB 0;"SOLUZI
ONI:" : PRINT
5260 FOR j=1 TO p
5262 IF CODE b$(h,j)=32 AND CODE
c$(h,j)=32 THEN GO TO 5275
5265 PRINT TAB 2;j;"")TAB 4;b$(
h,j):TAB 4;c$(h,j): PRINT
5270 NEXT j
5275 GO SUB 9600: GO TO 5800
5500 CLS
5502 LET p=VAL p$(h)
5505 FOR j=1 TO p
5510 IF u$(h,j)="s" THEN GO TO 5
5520 INVERSE 1: PRINT TAB 0;h;"")
"TAB 4;a$(h): INVERSE 0: PRINT
: PRINT TAB 1;"E":TAB 4;o$(h): P
RINT : PRINT TAB 0;j;"F":TAB 4;j
$(h,j): PRINT : PRINT TAB 1;"A":
TAB 4;u$(h,j)
5530 GO TO 5550
5540 INVERSE 1: PRINT TAB 0;h;"")
"TAB 4;a$(h): INVERSE 0: PRINT
: PRINT TAB 1;"E":TAB 4;o$(h): P
RINT : PRINT TAB 0;j;"F":TAB 4;j
$(h,j): PRINT : PRINT TAB 1;"A":
TAB 4;u$(h,j): PRINT : PRINT TAB
1;"D":TAB 4;t$(h): PRINT : PRIN
T TAB 1;"R":TAB 4;r$(h): PRINT :
PRINT TAB 1;"S":TAB 4;h$(h,j)
5550 GO SUB 9600
5560 NEXT j
5800 NEXT h
5850 GO TO 160

```

Correggere

Programma in BASIC (2000÷2400)

```

2000>REM CORREGGERE
2100 CLS : PRINT AT 10,0;"SCRIVI
IL NUMERO DELLA FRASE DA TAB 1
0" : CORREGGERE: GO SUB 9750
2105 LET f=1
2110 INPUT h
2112 IF h<=x THEN GO TO 2120
2115 CLS : PRINT AT 10,2;"LA FRA
SE "h;" NON E' INSERITA." : GO T
O 9550
2120 IF CODE a$(h)=32 THEN GO TO
2115
2125 CLS
2130 INVERSE 1 : PRINT TAB 0;h;"")
"TAB 4;a$(h): INVERSE 0
2132 FOR j=1 TO 4
2134 LET a$(h)="" : LET f$(h,1)=""
: LET f$(h,3)="" : LET f$(h,5)=""
: LET f$(h,6)="" : LET b$(h,j)=""
: LET c$(h,j)="" : LET j$(h,j)=""
: LET u$(h,j)="" : LET t$(h)=""
: LET r$(h)="" : LET h$(h,j)="" : L
ET p$(h)="" : LET o$(h)=""
2135 NEXT j
2140 GO TO 1110
2400 GO TO 160

```

Stampare

Programma in BASIC (6000÷6810)

```

6000>REM STAMPARE
6010 CLS
6020 FOR h=1 TO x

```



```

6030 IF CODE a$(h)=32 THEN GO TO
6000
6045 IF f$(h,1)="1" THEN GO TO 6
050
6047 IF f$(h,1)="2" THEN GO TO 6
250
6049 IF f$(h,1)="3" THEN GO TO 6
490
6050 INVERSE 1: LPRINT TAB 0;h;"
)"TAB 4;a$(h): INVERSE 0: LPRIN
T
6055 IF f$(h,5)="3" THEN LPRINT
TAB 4;m$(1): GO TO 6800
6060 IF f$(h,6)="1" THEN LPRINT
TAB 2;m$(2);TAB 6;m$(3)
6070 IF f$(h,6)="2" THEN LPRINT
TAB 2;m$(2);TAB 2;m$(4)
6080 IF f$(h,6)="3" THEN LPRINT
TAB 2;m$(2);TAB 1;m$(5)
6090 GO TO 6800
6250 INVERSE 1: LPRINT TAB 0;h;"
)"TAB 4;a$(h): INVERSE 0: LPRIN
T
6255 LPRINT : LPRINT TAB 0;"SOLU
ZIONI:" : LPRINT
6260 FOR J=1 TO P
6262 IF CODE b$(h,j)=32 AND CODE
c$(h,j)=32 THEN GO TO 6275
6265 LPRINT TAB 2;j;")"TAB 4;b$
(h,j);TAB 4;c$(h,j): LPRINT
6270 NEXT J
6275 GO TO 6800
6490 LPRINT
6500 CLS
6502 LET p=VAL P$(h)
6503 INVERSE 1: LPRINT TAB 0;h;"
)"TAB 4;a$(h): INVERSE 0: LPRIN
T : LPRINT TAB 0;"E":TAB 4;o$(h)
6505 FOR J=1 TO P
6510 IF u$(h,j)="s" THEN GO TO 6
540
6520 LPRINT : LPRINT TAB 0;j;"F"
)"TAB 4;j$(h,j): LPRINT : LPRINT
TAB 1;"A":TAB 4;u$(h,j)
6530 GO TO 6550
6540 LPRINT : LPRINT TAB 0;j;"F"
)"TAB 4;j$(h,j): LPRINT : LPRINT
TAB 1;"A":TAB 4;u$(h,j): LPRINT
: LPRINT TAB 1;"D":TAB 4;t$(h)
: LPRINT : LPRINT TAB 1;"R":TAB 4;
r$(h): LPRINT : LPRINT TAB 1;"S"
)"TAB 4;h$(h,j)
6550 LPRINT : LPRINT TAB 14;"***
*" : LPRINT
6560 NEXT J
6570 LPRINT
6800 NEXT h
6810 GO TO 180

```

Riportiamo di seguito l'elenco delle frasi da noi proposte per la verifica vs 1

1) Al cadere delle foglie inco-
mincia la primavera.

LA FRASE NON E' ACCETTABILE
PER DISACCORDO DI SIGNIFICATO

2) Ho detto a Giorgio che potr.
.. uscire dopo la partita.

SOLUZIONI:

- 1)potro'
io
- 2)potra'
Giorgio

3) Il mio pullover e' caldo an-
che se non e' di lana.

☐ non

10 Il mio non pullover e' caldo
anche se e' di lana.

☐ n

20 Il mio pullover non e' caldo
anche se e' di lana.

☐ s

5 QUALE TERMINE SI ACCORDA CON
non ?

☐ caldo

☐ n

3) Il mio pullover e' caldo non anche se e' di lana.

A) n

4) Mauro sulla libro ha letto un radioattivita'.

LA FRASE NON E' ACCETTABILE
PERCHE' DISORDINATA

5) Non sono uscita con Laura, perche' dovev... studiare.

SOLUZIONI:

1) dovevo
io

2) dovevamo
noi

3) doveva
Laura

6) Io e tua sorella potevamo andare al cinema.

LA FRASE E' ACCETTABILE

7) Questa mattina ho comprato la mia nuova agenda in quella libreria.

A) nuova

1) Questa mattina nuova ho comprato la mia agenda in quella libreria.

A) n

2) Questa mattina ho comprato la mia agenda in quella nuova libreria.

A) s

B) QUALE TERMINE SI ACCORDA CON
nuova?

C) libreria

S) n

3) Questa mattina ho nuova comprato la mia agenda in quella libreria.

A) n

8) Ho comprato i biscotti e li ho mangiato.

LA FRASE NON E' ACCETTABILE
PER DISACCORDO GRAMMATICALE

9) Quel bambino ha rotto un vaso cinese.

B) cinese

1) Quel bambino ha rotto cinese un vaso.

A) n

2) Quel bambino cinese ha rotto un vaso.

A) s

D) QUALE TERMINE SI ACCORDA CON
cinese?

B) bambino

S) n

10) Ho chiesto a Davide se poss. .. andare al cinema.

SOLUZIONI:

1) posso
io

2) possiamo
noi

Registrare

Programma in BASIC (9000÷9299)

```
9000>REM REGISTRARE
9005 CLS
9010 PRINT AT 10,0;"PER REGISTRA
RE CON REG.->PREMI 1"
9015 PRINT AT 12,0;"PER REGISTRA
RE CON MIC.->PREMI 2"
9020 GO SUB 9505
9022 IF x$="1" THEN GO TO 9100
9023 IF x$="2" THEN GO TO 9200
```

```

9024 BEEP 1,10: GO TO 9010
9100 SAVE "vs1" LINE 1
9199 GO TO 1
9200 ERASE "*" ; 1 ; "vs1"
9202 CLS : PRINT AT 9,3;"ORA REG
ISTRO LA VERIFICA."
9205 CLS : PLOT 20,92: DRAW +210
,0: DRAW 0,15: DRAW -210,0: DRAW
0,-15: PRINT AT 9,3;"ORA REGIST
RO LA VERIFICA."
9210 SAVE "*" ; 1 ; "vs1" LINE 1
9299 GO TO 1

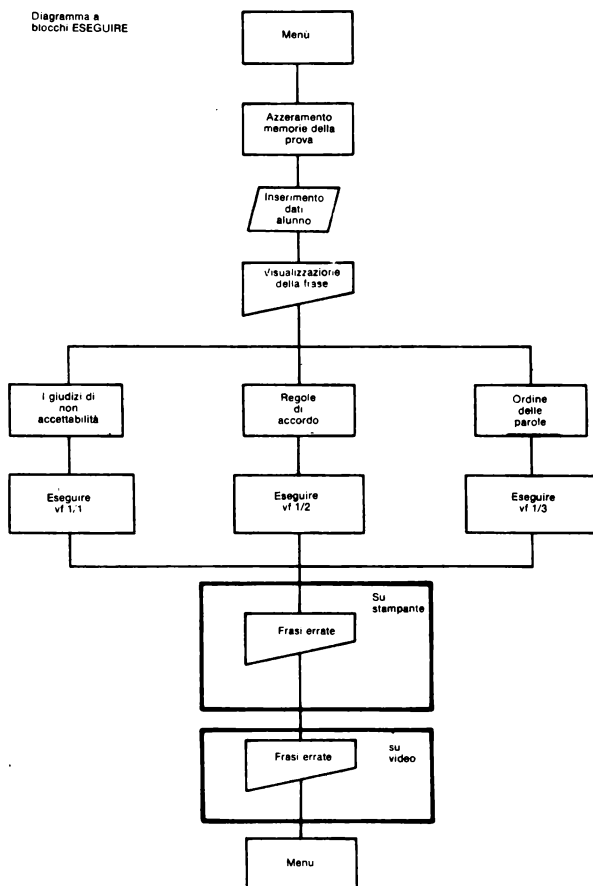
```

Eeguire

Le fasi operative del sottoprogramma ESEGUIRE risulteranno le stesse della vf 1/1:

- azzeramento delle memorie;
- memorizzazione dei dati dell'alunno;
- esecuzione della prova;
- stampa dei risultati della prova;
- visualizzazione dei risultati della prova.

Il diagramma a blocchi è:



Azzeramento delle memorie

Programma in BASIC (8000-8011).

```
8000>REM ESEGUIRE
8001 CLS
8002 FOR h=1 TO x
8003 IF CODE a$(h)=32 THEN GO TO
8005
8004 NEXT h
8005 LET s=h-1
8006 FOR h=1 TO x
8007 FOR j=1 TO 4: LET w$(h)=""
      LET f$(h,7)="" : LET f$(h,2)=""
      LET d$(h,j)="" : LET e$(h,j)=""
      LET i$="" : LET z$(h,j)="" : LET
8008 g$(h,j)="" : LET q$(h,j)="" : LET
      l$(h,j)="" : NEXT j
8008 NEXT h
8010 GO SUB 9710
8011 GO SUB 9600
```

Memorizzazione dei dati dell'alunno

Programma in BASIC (8013÷8036)

```
8013>PRINT AT 10,2;"SCRIVI IL TU
O COGNOME E NOME": GO SUB 9750
8015 INPUT i$(1 TO 24)
8016 CLS
8020 PRINT AT 10,7;"SCRIVI LA CL
ASSE": GO SUB 9750
8025 INPUT i$(26 TO 35)
8036 CLS
```

Esecuzione della prova

Lo svolgimento di questa parte del sottoprogramma sarà condizionato al tipo di giudizio espresso per ogni frase dall'insegnante.

L'alunno, infatti, si troverà direttamente a rispondere alle domande previste per quella frase, senza dover prima individuarne le caratteristiche.

Programma in Basic (8070÷8838)

```
8070>FOR h=1 TO s
8080 IF CODE a$(h)=32 THEN NEXT
h
8090 IF f$(h,1)="1" THEN GO TO 8
250
8100 IF f$(h,1)="2" THEN GO TO 8
500
8110 IF f$(h,1)="3" THEN GO TO 8
700
8250 REM GIUDIZI ACCETTABILITA'
8255 INVERSE 1: PRINT TAB 0,h;"
"TAB 4;a$(h): INVERSE 0: PRINT
8273 PRINT AT 10,3;#$(1): PRINT
AT 10,27;"?":TAB 13;"(s/n)"
8275 GO SUB 9500
8276 CLS
8278 LET f$(h,7)=x$
8280 IF f$(h,7)="s" THEN GO TO 8
302
8288 CLS: INVERSE 1: PRINT TAB
0,h;"":TAB 4;a$(h): INVERSE 0:
PRINT
8290 GO SUB 9650
8293 GO SUB 9505
8294 LET f$(h,2)=x$
```

```

8295 IF CODE f$(h,2)<>49 AND COD
E f$(h,2)<>50 AND CODE f$(h,2)<>
51 THEN BEEP 1,10: GO TO 8290
8297 IF f$(h,2)=f$(h,6) THEN GO
TO 8320
8300 LET w$(h)=a$(h)
8301 GO TO 8320
8302 IF f$(h,7)=f$(h,5) THEN GO
TO 8320
8310 LET w$(h)=a$(h)
8320 CLS : INVERSE 1: PRINT TAB
0;h;"":TAB 4;a$(h): INVERSE 0:
PRINT
8325 PRINT AT 10,3:"SEI SICURO D
ELLA RISPOSTA ?";TAB 13;"(s/n)"
8328 IF INKEY$<>"" THEN GO TO 83
25
8330 IF INKEY$="" THEN GO TO 833
0
8335 IF INKEY$="s" THEN GO TO 84
90
8338 LET f$(h,2)="" : LET f$(h,7)
=""
8340 CLS : GO TO 8255
8490 CLS : GO TO 8838
8500 REM ACCORDO SIGNIFICATO
8502 CLS
8505 INVERSE 1: PRINT TAB 0;h;"")
":TAB 4;a$(h): INVERSE 0
8510 PRINT AT 10,2:"QUANTE SOLUZ
IONI HA LA FRASE?"
8515 GO SUB 9505
8520 IF CODE x$(48 OR CODE x$>52
THEN GO TO 8515
8525 LET f$(h,4)=x$
8528 CLS
8530 LET v=VAL f$(h,4)
8540 FOR k=1 TO v
8545 INVERSE 1: PRINT AT 0,0;h;"
":TAB 4;a$(h): INVERSE 0: PRINT
AT 18,5:"SCRIVI LA ";k;" SOLUZI
ONE"
8550 GO SUB 9750
8555 INPUT d$(h,k)
8560 CLS
8565 GO SUB 9850
8570 PRINT AT 17,0:"SCRIVI L'ELE
MENTO DELLA FRASE";TAB 5;"CHE SI
ACCORDA CON LA ";TAB 10;k;" SOL
UZIONE"
8575 GO SUB 9750
8580 INPUT e$(h,k)
8585 CLS
8588 GO SUB 9850
8590 NEXT k
8603 PRINT AT 18,3:"SEI SICURO D
ELLE RISPOSTE ?";TAB 13;"(s/n)"
8610 GO SUB 9500
8615 IF x$="s" THEN GO TO 8630
8620 FOR j=1 TO 4: LET d$(h,j)=""
": LET e$(h,j)="" : NEXT j
8625 CLS : GO TO 8505
8630 CLS : LET p=VAL f$(h,3): LE
T v=VAL f$(h,4)
8635 FOR j=1 TO p
8640 FOR k=1 TO v
8645 IF b$(h,j)=d$(h,k) AND c$(h
,j)=e$(h,k) THEN LET y$(h,k)=d$(
h,k): LET z$(h,k)=e$(h,k): LET d
$(h,k)=d$(h,j): LET e$(h,k)=e$(h
,j): LET d$(h,j)=y$(h,k): LET e$(
h,j)=z$(h,k): GO TO 8680
8650 NEXT k
8670 LET w$(h)=a$(h)
8680 NEXT j
8685 CLS : GO TO 8838
8700 REM ESEGUIRE ORDINE PAROLE
8702 CLS
8705 LET p=VAL p$(h)
8710 INVERSE 1: PRINT TAB 0;h;"")
":TAB 4;a$(h): INVERSE 0: PRINT
": INVERSE 1: PRINT TAB 0;"FRASI:
": INVERSE 0: PRINT
8715 FOR j=1 TO p: PRINT TAB 2;j
":TAB 4;j$(h,j): PRINT : NEXT
j
8720 GO SUB 9500
8725 CLS
8730 FOR j=1 TO p
8735 INVERSE 1: PRINT j;"":TAB

```

```

4:J$(h,j): INVERSE 0: PRINT AT 1
0,3;"LA FRASE E' ACCETTABILE?";
TAB 13;"(s/n)"
8740 GO SUB 9500
8742 CLS
8745 LET g$(h,j)=x$
8747 NEXT J
8750 INVERSE 1: PRINT TAB 0;h;"
";TAB 4,3$(h): INVERSE 0: PRINT
"; INVERSE 1: PRINT TAB 0;"FRASI:
"; INVERSE 0: PRINT
8750 FOR J=1 TO P
8755 PRINT TAB 0;J;"");TAB 4;J$(
h,j)
8760 NEXT J
8765 PRINT AT 16,1;"CONFRONTANDO
LA FRASE BASE CON";TAB 3;"LE FR
ASI SUCCESSIVE QUAL'E";TAB 5;"L
ELEMENTO SPOSTATO?"
8767 GO SUB 9750
8790 INPUT n$(h)
8792 IF o$(h)<n$(h) THEN GO TO
8820
8795 CLS
8797 FOR J=1 TO P
8800 IF u$(h,j)="s" THEN INVERSE
1: PRINT TAB 0;J;"");TAB 4;J$(h
,j): INVERSE 0
8802 NEXT J
8805 PRINT AT 10,1;"D";TAB 4;t$(
h)
8807 GO SUB 9750
8810 INPUT q$(h)
8812 CLS
8815 INVERSE 1: PRINT TAB 0;h;"
";TAB 4,3$(h): INVERSE 0: PRINT
"; INVERSE 1: PRINT TAB 0;"FRASI:
"; INVERSE 0: PRINT
8817 FOR J=1 TO P
8818 IF u$(h,j)="s" THEN PRINT T
AB 0;J;"");TAB 4;J$(h,j): GO TO
8822
8820 NEXT J
8822 PRINT AT 18,4;"LE FRASI HAN
NO LO STESSO";TAB 8;"SIGNIFICAT
O";TAB 22;"(s/n)"
8825 GO SUB 9500
8827 LET l$(h,j)=x$
8828 CLS
8829 PRINT AT 12,2;"SEI SICURO D
ELLE RISPOSTE?";TAB 13;"(s/n)"
8830 GO SUB 9500
8831 IF x$="s" THEN GO TO 8834
8832 FOR J=1 TO 4: LET n$(h)=""
LET g$(h,j)="" : LET q$(h)="" : L
ET l$(h,j)="" : NEXT J
8833 CLS : GO TO 8705
8834 FOR J=1 TO P
8835 IF o$(h)<n$(h) OR u$(h,j)<
>q$(h,j) OR r$(h)<>q$(h) OR h$(h
,j)<>l$(h,j) THEN LET w$(h)=a$(h
,j)
8837 NEXT J
8839 CLS : NEXT h

```

Il programma si presenta diviso in tre parti ognuna delle quali contiene le stesse linee di istruzioni utilizzate per l'ESECUZIONE DELLA PROVA di ogni singola verifica.

Il riconoscimento della frase e delle sue caratteristiche è dato dalle condizioni relative alla memoria f\$(h, 1); questo indirizzo di memoria, contenendo le risposte dell'insegnante relative ai dati della frase, determinerà lo svolgimento di una delle tre parti del programma (8250 ACCETTABILITÀ — 8500 REGOLE DI ACCORDO — 8700 ORDINE DELLE PAROLE)

Durante la prova le risposte fornite dall'alunno verranno confrontate con quelle inserite dall'insegnante e la presenza di un solo errore determinerà la memorizzazione della frase esaminata.

Stampa dei risultati della prova

Programma in BASIC (8840÷8870)

```

8840>LPRINT AT 1,0;"ALUNNO: ";i$(
1 TO 24); LPRINT : LPRINT "CLAS
SE: ";i$(26 TO 35)
8841 LET n=0
8842 FOR h=1 TO s
8845 IF CODE w$(h)=32 THEN GO TO
8848
8846 IF CODE w$(h)<>32 THEN LET
n=n+1
8848 NEXT h
8850 IF n=0 THEN LPRINT : LPRINT
AT 5,0;" NESSUNA FRASE DA CONT
ROLLARE "; GO TO 8875
8853 LPRINT : LPRINT AT 5,0;" CO
NTROLLARE LE FRASI SEGNALATE ";
LPRINT
8853 FOR h=1 TO s
8854 IF CODE w$(h)=32 THEN GO TO
8870
8855 INVERSE 1: LPRINT TAB 0;h;"
";TAB 4;w$(h); INVERSE 0
8860 LPRINT : LPRINT TAB 14;"***
*
8870 NEXT h

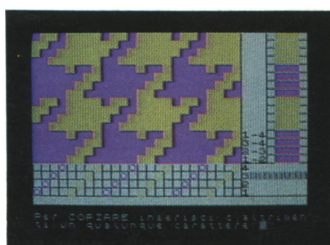
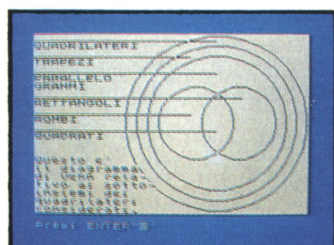
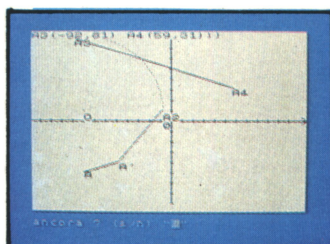
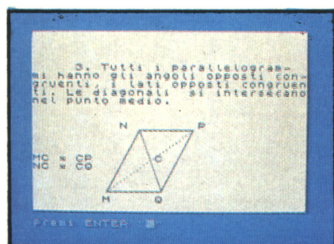
```

Visualizzazione dei risultati della prova **Programma in BASIC (8875÷8999)**

```

8875>PRINT AT 1,0;"ALUNNO: ";i$(
1 TO 24); PRINT : PRINT "CLASSE:
";i$(26 TO 35)
8878 LET n=0
8880 FOR h=1 TO s
8883 IF CODE w$(h)=32 THEN GO TO
8886
8885 IF CODE w$(h)<>32 THEN LET
n=n+1
8886 NEXT h
8890 IF n=0 THEN PRINT AT 5,0;"
NESSUNA FRASE DA CONTROLLARE "
GO TO 8990
8893 PRINT : PRINT AT 5,0;" CONT
ROLLARE LE FRASI SEGNALATE "; PR
INT
8894 FOR h=1 TO s
8895 IF CODE w$(h)=32 THEN NEXT
h
8900 INVERSE 1: PRINT TAB 0;h;"
";TAB 4;w$(h); INVERSE 0
8910 PRINT : PRINT TAB 14;"****"
PRINT
8960 NEXT h
8990 GO SUB 9600
8999 GO TO 1

```

RISORSE PER LA DIDATTICA è una delle prime agenzie di aggiornamento del personale docente della scuola italiana ad aver creduto all'informatica come strumento efficace di rinnovamento dell'azione pedagogico-didattica.

Opera nel settore da tempo ed ha certamente contribuito a far conoscere i processi e i prodotti dell'uso del computer nell'ambiente degli insegnanti attraverso il suo primo progetto "Elaborazione in classe". Dopo i primi anni dedicati soprattutto alla fase di "alfabetizzazione informatica" ha ritenuto di dover contribuire all'evoluzione dei modelli di insegnamento attraverso il computer.

Questa pubblicazione è proprio il frutto di una prima messa in atto della seconda fase della sua "strategia": quella rivolta al problema dell'uso e della produzione del software didattico.

Il progetto di questa seconda fase, "Banca di UDS (Unità Didattiche Software)", fu messo in cantiere dopo un affollato seminario ai margini della mostra sul libro scientifico e tecnico, allestita alla fiera di Milano dall'amministrazione provinciale nel 1984.

Si può tentare un insegnamento che integri coi vari sistemi già presenti nella scuola anche quello dell'uso del computer? In tutte le discipline? Anche nella scuola dell'obbligo?

È legittimo un insegnamento dell'uso del computer e dei linguaggi artificiali nella scuola di base? Contribuisce alla formazione e all'orientamento degli alunni in una scuola che, per la sua natura, non deve essere "professionalizzante"?

È possibile mettere i docenti in condizione di non dover subire il software didattico "a scatola chiusa", ma di saperlo leggere, interpretare, trasformare e utilizzare flessibilmente così come essi hanno imparato a fare con i libri di testo?

È possibile una formazione del personale docente che favorisca la produzione di software finalizzato alla scuola o come minimo alla "personalizzazione" del software esistente?

Già le prime risposte stanno arrivando e questo libro ha il pregio di raccoglierle, renderle comprensibili attraverso esempi concreti e inquadrarle in una enunciazione di strategie possibili.

Gli esempi presentati, arricchiti con quasi duecento disegni e sessanta programmi, sono spiegati in modo da rendere chiaro l'uso del software ma anche da rendere possibile la sua modificazione.

173

Dal libro di testo all'uso dello Spectrum

Risorse per la Didattica

