

ORE 10: LEZIONE DI BASIC

Corso completo di programmazione Basic in 21 lezioni sul
COMMODORE C64

di AMADIO GOZZI



Ricordiamo che ogni ciclo si chiude con la parola NEXT. Le istruzioni che
sono fornite fra FOR e NEXT, vengono ripetute a ogni ciclo. Es. se fra
FOR e NEXT vengono dati in successione le istruzioni PRINT K e PRINT
di ogni passaggio viene scritto il valore della variabile K assunto in quel
momento e creato uno spazio in verticale prima di scrivere il valore successivo.
UTILIZZAZIONE DEL FOR...NEXT
elemento in verticale
do debbono essere delle spaziatrici verticali di diverse dimensioni, al
di scrivere tanti PRINT, si può usare l'espressione:
FOR K = 1 TO N : PRINT : NEXT K
è il numero delle inserienze desiderato.
l'espressione
separare due parti di un elaborato con una linea bianca continua
si, si può usare l'espressione:
L = 1 TO 40 : PRINT "E" : NEXT K
è il segno grafico che si ottiene premendo (dopo aver inserito
SHIFT) la lettera, C. Diciamo per inciso che se si preme il tasto
al posto di SHIFT, invece del segno □ viene stampato quello
—STEP
razione ciclica a passo". In questo caso, la variabile non sale
ti, ma seguendo dei salti regolari. Es.:
I TO 20 STEP 2
o del ciclo, K varia di 2 unità. In questo caso specifico, K
di 1-3-5-7-9-11-13-15-17-19.
EP—
e ciclica inverte, cioè il com-
no per ripetere "

Commento all'esercizio n. 9
Le righe 70-90 rappresentano una normale operazione ciclica di conteggio da
1 a 20 con stampa in orizzontale dei valori assunti via via dalla variabile K.
Le righe 130-150, ripropongono l'operazione ciclica precedente, ma all'interno del
LOOP o CICLO, viene eseguita l'espressione S=S+K che serve a fare la
somma dei tutti i valori assunti da K nei 10 cicli (vedere la spiegazione
nell'Appendice 2) (Fig. 6).

FOR K = 1 TO 20
NEXT K
(Chiuso il ciclo)

Fig. 6

FOR K = 1 TO 20
NEXT S
(Chiuso il ciclo)

S = S + K
(Per la somma)

Ad ogni ciclo, la variabile S si incrementa del valore
ciclo stesso. Così facendo, S rappresenta
che K ha assunto in precedenza
Per K = 1 S = 1
Per "

edizioni **Jce**

LIBRO + CASSETTA

ORE 10: LEZIONE DI BASIC

**Corso completo di programmazione Basic in 21 lezioni sul
COMMODORE C64**

di AMADIO GOZZI



**JACOPO CASTELFRANCHI EDITORE
Via dei Lavoratori, 124
CINISELLO BALSAMO (MI)**

Tutti i diritti sono riservati, nessuna parte di questo libro e della cassetta software allegata possono essere riprodotti, posti in sistemi di archiviazione, trasmessi in qualsiasi forma o mezzo elettronico, meccanico, di fotocopiatura, ecc., senza l'autorizzazione scritta dell'Editore.

Nel testo sono stati introdotti programmi di valore didattico. L'Editore non risponde dei possibili errori che si verifichino nei listati e nei relativi risultati.

Prima edizione: Giugno 1985

Pubblicato in Italia da:

JCE Jacopo Castelfranchi Editore
Via dei Lavoratori, 124
20092 Cinisello Balsamo (MI)

Copyright © Amadio Gozzi, 1985

Sistema di fotocomposizione elettronica: JCE

Stampato da:

Gemm Grafica S.r.l.
Via Magretti, Paderno Dugnano (MI)

INDICE GENERALE

| | Pagina |
|---|--------|
| Prefazione | 3 |
| Premessa | 6 |
| Lezione 1 - PRINT - RETURN - LIST - RUN - SHIFT/CLR | 7 |
| Lezione 2 - Il bit - RAM e ROM - Il byte - NEW - PRINT:PRINT - SAVE | 11 |
| Lezione 3 - Interfaccia - Trasmissione dati - Modem - Office - Automation - TAB(X) - SHIFT+INST | 15 |
| Lezione 4 - Hardware - Software - Basic - Assembler - LOAD - INPUT -REM - RUN STOP - RESTORE | 19 |
| Lezione 5 - Stampante - Grafica - ALU - INPUT "..."; LU | 25 |
| Lezione 6 - Vari usi di PRINT - DEL - INST+SHIFT - IF...THEN -GOTO - \$ (Stringa) - Azzeramento variabili | 29 |
| Lezione 7 - PRINT "☑" - R\$ (variabile stringa) - CRSR - Scelta logica con INPUT - Diverso < > - GET | 33 |
| Lezione 8 - FOR...NEXT - FOR...NEXT...STEP (FOR..NEXT...STEP-) - Clock Multiprogrammazione - BUS Sommatore S = S+K | 37 |
| Lezione 9 - GOSUB - END | 41 |
| Lezione 10 - GOSUB | 45 |
| Lezione 11 - Linguaggi - Memoria RAM - Indirizzamento - Capacità di memoria - ON...GOTO | 49 |
| Lezione 12 - READ...DATA (Aperto) - READ...DATA (Chiuso) con FOR...NEXT - SPC(X) | 53 |
| Lezione 13 - Personal Computer - Word Processing - Il Triumph Adler P2 -Caricamento programma - Inizializzazione dischetto | 57 |
| Lezione 14 - Operazione ciclica doppia (Doppio Loop) - Calcolo del numero delle ripetizioni - Formule | 61 |
| Lezione 15 - Matrici lineari - Salvataggio dati su disco | 65 |
| Lezione 16 - Creazione della maschera - Ritorno in alto e riposizionamento del cursore per gli input | 69 |
| Lezione 17 - Esercizio finale n. 21 - Creazione indice (menù) e subroutines | 73 |
| Lezione 18 - Problemi nell'osservazione del Listing - CONTROL - Esercizio finale n. 21 - Sottoprogramma n. 1 - INTRODUZIONE DATI - Gestione cassette | 79 |
| Lezione 19 - Esercizio finale n. 21 - Sottoprogramma n. 2 - CONSULTAZIONE DATI | 83 |

| | Pagina |
|--|---------------|
| Lezione 20 - Esercizio finale n. 21 - Sottoprogramma n. 3 | |
| AGGIORNAMENTO DATI - STAMPA DATI - | 85 |
| Listato esercizio n. 21 | 89 |
| Appendice 1 - Riassunto funzioni INPUT-RUN-LIST-PRINT | 95 |
| Appendice 2 - Salvataggio dati su cassetta e su disco | |
| - Recupero dati - Stampa del LIST e del RUN. List e | |
| Riordino dati - Sommatoria di dati | 99 |
| Appendice 3 - Correzione | 103 |
| Appendice 4 - Funzioni FRE(X) - RND - Input in linea | 105 |
| Appendice 5 - Indice degli esercizi della cassetta | 107 |
| List e lettura esercizio n. 1 | 109 |
| List e lettura esercizio n. 2 | 110 |
| List e lettura esercizio n. 3 | 111 |
| List e lettura esercizio n. 4 | 112 |
| List e lettura esercizio n. 5 | 113 |
| List e lettura esercizio n. 6 | 114 |
| List e lettura esercizio n. 7 | 115 |
| List e lettura esercizio n. 8 | 116 |
| List e lettura esercizio n. 9 | 118 |
| List e lettura esercizio n. 10 | 119 |
| List e lettura esercizio n. 11 | 121 |
| List e lettura esercizio n. 12 | 123 |
| List e lettura esercizio n. 13 | 124 |
| List e lettura esercizio n. 14 | 125 |
| List e lettura esercizio n. 15 | 126 |
| List e lettura esercizio n. 16 | 127 |
| List e lettura esercizio n. 17 | 128 |
| List e lettura esercizio n. 18 | 130 |
| List e lettura esercizio n. 19 | 131 |
| List e lettura esercizio n. 20 | 133 |
| Indice istruzioni e comandi | 135 |
| Indice argomenti di informatica | 136 |

PREFAZIONE

Si voglia o no, la maggior parte del materiale divulgativo di base sull'uso creativo dei computer è di provenienza straniera. Lo dimostrano, tra l'altro, le frasi inserite nei listati, che nella fretta di dover arrivare ad ogni costo prima degli altri, sono rimaste in lingua originale tradendo in tal modo la loro origine.

L'introduzione dei calcolatori nel nostro paese, così come è avvenuto per i televisori a colori, per i videoregistratori e per altre apparecchiature elettroniche ad altissima integrazione, è avvenuta con una operazione di tipo quasi esclusivamente commerciale. Nessuna industria italiana, al momento, progetta calcolatori, se per progettare s'intende creare dall'A alla Z sia i componenti di base sia i circuiti che dovranno utilizzarli.

Ne deriva il fatto che l'Italia oggi, e lo sarà ancora per molto tempo, è di fatto invasa da decine di multinazionali dell'informatica che a suon di miliardi si contendono un mercato che diventa sempre più lauto di giorno in giorno. Ne consegue che in pochi anni, centinaia di migliaia di apparecchiature assai sofisticate verranno messe in mano ad altrettante centinaia di migliaia di operatori che di questa materia non sanno molto di più che premere una decina di tasti di prima utilità. Come dire che, parafrasando una frase celebre, fatti i calcolatori ora occorre fare degli italiani informatici. Il che è tutta un'altra cosa.

Già si è partiti col piede sbagliato. Le aziende commerciali, per introdurre più velocemente i computer in circolazione (non ha importanza quali e come e dove, basta vendere), li presentano come dei giochini e accompagnano la vendita con una miriade di programmi che i fruitori copiano pappagallescamente fino al momento in cui non sopravviene la stanchezza o non cambia la moda e li abbandonano nel solito cantuccio o nella solita mansarda. Il computer non è stato ancora presentato adeguatamente come un vero strumento di lavoro e di potenziamento delle proprie capacità ideative per tutti i cittadini e quindi come uno strumento di miglioramento collettivo. Al momento anzi, per molte categorie di persone esso viene visto soltanto come un generatore di disoccupazione.

Questo primo libro redatto dalla Basic School ai cui corsi si informa, è uno dei primi tentativi di "italianizzare" la cultura del computer e di sovvertire il corso convulso e non sempre positivo fin qui seguito dalla divulgazione in questo campo.

ORE 10 LEZIONE DI BASIC è tratto infatti dai corsi di base della BASIC SCHOOL. Si tratta di corsi che durano 22 ore complessive durante il quale, qualsiasi cittadino che abbia una cultura media, di qualunque età o sesso, con diverse esperienze di studio e di lavoro, impara a programmare in linguaggio BASIC.

Più avanti verranno pubblicati altri libri simili, per il perfezionamento del Basic, per l'acquisizione di altri linguaggi come il COBOL, il CP/M ecc.

Il corso si suddivide in una ventina di lezioni, di 1 ora - 1 ora e mezza ciascuna. È particolarmente adatto per l'insegnamento del Basic nelle scuole medie inferiori e superiori.

È strutturato in modo da favorire anche l'apprendimento autodidatta, magari dilatando un poco il numero delle lezioni.

È importante che non ci si lasci prendere dalla frenesia che solitamente accompagna l'operatore quando incomincia a comprendere che il computer gli offre possibilità di ideazione praticamente illimitata. È bene seguire pedissequamente le lezioni senza saltare di qua e di là.

Imparare è come salire su una scala. Saltare dei gradini lascia delle lacune nella preparazione ed ingenera confusione.

Consigliamo anche di non applicarsi per più di un'ora al giorno almeno fino a quando non si sia arrivati in fondo a questo primo corso di base.

Un'ora al giorno per 5 giorni massimo alla settimana, riposando il sabato e la domenica, si sono nella pratica dimostrati ottimali per lasciar tempo al cervello di incamerare senza fatica tutte le decine di istruzioni e di concetti di cui si compone il Basic Interpreter.

Per quanto riguarda l'impiego di questo libro nella scuola, lo stesso modo con cui è stato compilato sta a dimostrarne l'intento prettamente didattico. Niente giochini, palline che rimbalzano, guerre stellari o simili, ma un rigoroso progredire tra teoria informatica e pratica applicativa, privilegiando all'inizio la prima e verso il termine la seconda, con un andamento si può dire naturale dell'esperienza didattica.

All'inizio infatti, occorre fornire alcuni concetti teorici, perchè l'allievo si senta a poco a poco immerso nella dimensione informatica. Gli esercizi sono molto facili e per qualche giorno la pratica è più ripetitiva che concettuale, anche per favorire una più rapida diteggiatura e una presa di contatto più completa con la molteplice realtà della tastiera. A partire dalle lezioni 7-8, la parte teorica scende come importanza, mentre il lavoro di gestione dei mini-programmi acquista più spazio, diventa più intenso, con una velocità esecutiva che va in progressione, per sfociare poi nella gestione dell'esercizio-saggio finale, il n. 21 il quale contiene in sé praticamente tutto quanto acquisito in precedenza e inoltre ha la struttura portante di molti programmi di ben più ampia mole.

Questo corso di Basic è nato sul computer Commodore 64, il quale dal punto di vista didattico offre possibilità quasi perfette. Esso però si può adattare a qualsiasi altro computer che sia almeno di pari importanza come versatilità. Laddove, nei listati esistano delle soluzioni che sono tipiche del Commodore, vengono suggerite delle alternative per chi possiede altri tipi di calcolatori. Per quanto riguarda la strutturazione del libro è bene precisare che:

- a - Ogni lezione comprende cinque sezioni: Lezione pratica - Notizie d'informatica - Fotografia del programma eseguito con spiegazione - Listato dell'esercizio con lettura dello stesso - Programma su nastro per avere la possibilità di caricarlo e vedere come gira il RUN (esecuzione).*
- b - La lezione teorica sull'informatica potrà precedere o seguire la parte*

- pratica. La nostra esperienza ci suggerisce di svolgerla all'inizio della lezione, perchè gli allievi sono poco propensi, una volta iniziato, ad interrompere l'esercizio pratico che li appassiona anche forse in eccesso.*
- c - Dopo le prime lezioni, in cui occorre in continuazione raccordare RUN e LIST per creare una corrispondenza biunivoca fra causa ed effetto, è bene che il LIST sia letto il meno possibile e che si cerchi di programmare osservando l'esecutivo. A questo scopo, quando trattasi di una scuola, occorre che l'insegnante abbia un computer a sua disposizione collegato con un monitor (basta un televisore in bianco e nero) su cui proiettare i programmi eseguiti per farli interpretare e tradurre in linee di programma dagli allievi.*
- d - Per chi voglia apprendere autodidatticamente a programmare consigliamo di utilizzare il LISTING il meno possibile e soltanto quando dall'esecutivo non traspaia chiaramente l'intento di chi ha preparato l'esercizio.*

Ringraziamenti

Al termine di questa lunga chiacchierata che per metà è prefazione e per metà è spiegazione tecnica, mi è cosa gradita ringraziare una persona che ha dato il suo valido contributo alla realizzazione di un'opera così nuova e solo all'apparenza semplice.

Ringrazio perciò il Signor:

Paolo Colombo per la supervisione tecnica dell'intera opera

Ringrazio pure tutti quei lettori, insegnanti e non, che con critiche e suggerimenti mi vorranno scrivere per poter migliorare e rendere più incisivo il lavoro didattico alle prossime edizioni. Potranno scrivere alla

Basic School

c/o J.C.E.

Via dei Lavoratori 124

20092 Cinisello B. (MI)

Tel. (02) 61.72.641

Buon lavoro a tutti.

L'autore

PREMESSA

Specialmente nelle prime lezioni, agli esercizi pratici di programmazione su computer, si accompagnano delle notizie di informatica. Ciò per fornire all'allievo il necessario supporto culturale per entrare a pieno titolo nel mondo dei calcolatori. All'inizio sarà bene che le notizie precedano l'esercizio pratico, anche perchè l'allievo si stancherebbe nello stare troppo a lungo alla tastiera. Dopo la metà del corso, invece, la parte pratica diventerà preponderante a scapito della teoria.

N.B. In fase di programmazione, l'operatore (il lettore) si rivolge al computer con il TU, come se gli desse degli ordini.



Tastiera di computer Commodore 64. Si tratta di un calcolatore molto versatile e con un altissimo rendimento didattico. Ha una buona possibilità grafica per una risposta anche estetica dei programmi. Inoltre, se munito di monitor o TVC, si può giocare anche con i colori per rendere più gradevole il PRINTATO. Può essere impiegato anche per fare musica. Può trasformarsi da Home a Personal Computer se lo si collega ad un monitor semiprofessionale, ad un paio di lettori di floppy-disk da 5" e ad una stampante adeguata.

BASIC SCHOOL - Lezione n. 1

Lezione pratica

Quello che viene definito il "Ciclo operativo primario" col quale si inizia questo corso, comprende alcune istruzioni ben definite che di solito vengono ripetute nella stessa sequenza:

- PRINT** Fai partire il cursore da destra verso sinistra. Se durante il percorso incontri una frase fra gli apici: "FRASE" oppure una variabile (P oppure P\$), scrivi sul video la frase o la variabile, altrimenti torna a capo spaziando 1 interlinea.
- RETURN** (tasto). Immagazzina nella tua memoria RAM la istruzione o le istruzioni ricevute in precedenza. È forse il tasto più utilizzato e va premuto ogniqualvolta si vuole rendere accettata dal calcolatore una riga di programma o una istruzione o un comando oppure una correzione sul programma stesso.
- SHIFT+CLR** Ripulisci lo schermo del video e riporta il cursore in alto a sinistra.
- RUN** Dài corso al programma precedentemente impostato.
- LIST** Fai vedere la lista del programma.
- 10** Riga di programma.
- 20** Riga di programma.
- 30** Riga di programma.

NOTE

È indispensabile ripulire lo schermo (Tasti SHIFT e CLEAR insieme) ogni qualvolta si passa dal LIST al RUN.

Il **cursore** (quadratino luminoso pulsante) equivale alla pallina della macchina da scrivere elettrica che si sposta su carrello fisso.

Il tasto SHIFT assomiglia al tasto della maiuscole di una macchina da scrivere. Nei tasti che hanno due caratteri (grafica esclusa) (es.: 2 e "), lo SHIFT serve per battere quello superiore. Quando una riga di programma contiene errori, è meglio rifarla, riscrivendola in modo esatto con lo stesso numero indicativo. Dopo il RETURN, viene automaticamente cancellata quella precedente.

Le righe vanno numerate di 10 in 10 per potervi all'occorrenza inserire delle correzioni. Attenzione a non scambiare la lettera O per lo 0 (zero). In questo caso occorre cancellare la riga male impostata. Es.: 2 O PRINT "BASIC SCHOOL". Il computer la considererà come la riga n. 2 e non 20. Per correggere, battere 2 e RETURN (così si cancella) e rifare la riga di programma impostando il numero esatto (20).

Per il momento viene utilizzata la tabulazione automatica che corrisponde a 10 battute normali e che viene attuata scrivendo una virgola dopo l'istruzione PRINT. Es.:

10 PRINT, "BASIC SCHOOL 1"

La frase BASIC SCHOOL resta tabulata in modo automatico di 10 battute dal margine sinistro

10 PRINT,, "BASIC SCHOOL 1"

La frase BASIC SCHOOL resta tabulata automaticamente di 20 battute, ecc.

Quando l'istruzione PRINT non è seguita da niente, significa che il cursore percorre una riga e poi ritorna a capo, un'interlinea sotto. In sostanza si spazia in verticale di 1 interlinea.

Per spostare il cursore senza cancellare i caratteri, servirsi di **CRSR →** e **CRSR ↑** Con lo SHIFT lo si sposta in alto e a sinistra. Senza lo SHIFT, in basso e a destra.

Per cambiare un carattere sbagliato, se si è in fase di scrittura del programma, si torna indietro, cancellando, con il tasto DEL e si ribattono i caratteri sbagliati.

Quando la riga sia già in memoria (RETURN battuto), si ritorna con i due CRSR sul carattere sbagliato, si batte il carattere giusto e poi si preme RETURN per memorizzare la correzione; quindi si esce in basso con il CRSR per portare il cursore oltre il READY.

Il READY è un messaggio attraverso il quale il calcolatore si dichiara pronto a ricevere comandi o istruzioni.



Esercizio n. 1
RUN (esecuzione).

Esercizio n. 1

(Vedere il RUN n. 1, cioè come il programma appare sullo schermo dopo l'esecuzione).

Scrivere sullo schermo:

```
10 PRINT "BASIC SCHOOL 1"  
40 PRINT "COGNOME E NOME"
```

quindi correggere spaziando 2 interlinee in verticale tra le due frasi

```
20 PRINT  
30 PRINT
```

Completare l'esercizio tenendo conto delle tabulazioni automatiche 1 e 2 (PRINT , "FRASE" oppure PRINT,, "FRASE") e delle spaziature in verticale.

Notizie d'informatica

La configurazione delle apparecchiature che fanno parte di una stazione elaboratrice è rappresentata in fig. 1.

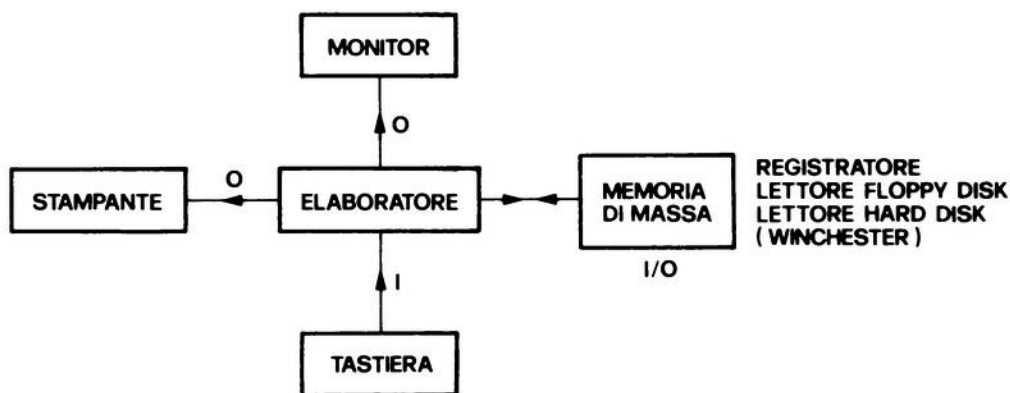


Fig. 1

Come si può vedere, l'elaboratore centrale (la dicitura "calcolatore" è troppo riduttiva, in quanto questa macchina più che far calcoli, compito che per altro svolge egregiamente, elabora dei programmi inseriti da un operatore esterno) ha applicate alla sua periferia delle apparecchiature ausiliarie chiamate anche I/O ovvero Input/Output (Ingresso/Uscita). Queste aggiunte sono:

1. *Tastiera*. È una apparecchiatura di Input in quanto serve a dialogare tra l'operatore e l'elaboratore, introducendo in esso istruzioni di programma oppure dati, sia letterali che numerici.
2. *Stampante*. Si tratta di una Output. Infatti i dati elettrici escono dal centro verso di essa per poter stampare su carta dei LISTING oppure dei programmi finiti.

3. *Monitor Video*. Anch'esso è un Output. I dati vanno dal calcolatore allo schermo di visione. È un elemento di controllo di quanto viene svolto sia dal programmatore che dall'elaboratore stesso.
4. *Memoria di massa*. È un I/O in quanto i dati passano dal centro alla cassetta oppure al Floppy Disk quando si vogliono salvare dati e/o programmi. Al contrario i dati passano dalla memoria di massa al calcolatore quando si intendano caricare nella memoria RAM dell'elaboratore il programma e/o i dati numerici o letterali che fanno parte dell'esecuzione del programma stesso. Va infatti ricordato che la memoria RAM del calcolatore, quando questo viene spento, perde sia i programmi di lavoro che i dati.

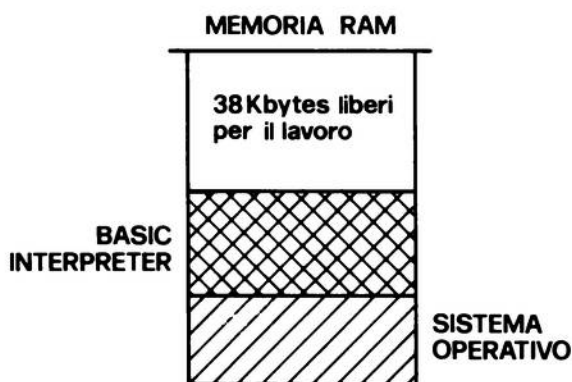


Fig. 2

Presentazione del Commodore 64

Si tratta di un elaboratore assai versatile che permette di svolgere una grande quantità di programmi di piccola e media mole. Oltre a ciò dispone di molte possibilità per quanto riguarda i disegni grafici e il trattamento dei colori, quando venga collegato con un monitor a colori oppure ad un TVC.

All'accensione presenta l'espressione - *38911 bytes free RAM*. Significa che di tutta la sua capacità di memoria interna che è di 64 k bytes (vedi figura 2), 38911 bytes sono disponibili per i programmi e i dati (capacità di lavoro), mentre il resto viene coperto dal Sistema Operativo interno e dall'interprete BASIC (Fig. 2).

Quest'ultimo è contenuto in maniera fissa in una apposita memoria a sola lettura chiamata ROM. Il sistema operativo (S.O.) è un insieme di istruzioni fisse interne al calcolatore che servono alla gestione dei diversi componenti della macchina (gestione del monitor, della memoria, ecc.).

BASIC SCHOOL - Lezione n. 2

Notizie di informatica

IL BIT

Il bit è l'informazione elettrica elementare su cui si basa tutta la cosiddetta elettronica digitale. Consta di due stati elettrici ben precisi: presenza di tensione (Es.: 5 Volt) o stato Alto e assenza di tensione (Es.: 0 Volt) o stato Basso. Simbolicamente, il bit si indica con **1** quando è nello stato alto e con uno **0** (zero) quando è nello stato basso.

Il bit può essere paragonato alla informazione luminosa che si ricava da una torcia che può trovarsi accesa (bit 1) oppure spenta (bit 0). Combinando più bit del tipo 1 e del tipo 0, si possono creare delle parole in codice. Le combinazioni possibili sono fornite dalla formula, per n bit:

$$\text{Combinazioni} = 2^n$$

Così ad esempio, con 2 bit possiamo comporre $2^2 = 4$ parole in codice così combinate 00 - 11 - 01 - 10.

IL BYTE

Il byte è una parola in codice binario (cioè composta di bit) che si avvale di 8 bit disposti uno di fianco all'altro. Le combinazioni che si ottengono con 1 byte sono $2^8 = 256$. Nei calcolatori a 8 bit, ogni qualvolta viene immesso un'insieme contemporaneo di 8 bit (cioè 1 byte), essi camminano su 8 collegamenti che trasporteranno 1 byte alla volta (vedi BUS dei dati).

CALCOLATORI A 8-16-32 BIT

Naturalmente vi sono istruzioni che contemplano l'invio di più bytes. Per cui, sono stati progettati microprocessori coi quali sono stati costruiti calcolatori a 2 bytes (16 bit) e 4 bytes (32 bit). Questi ultimi sono caratterizzati da una più alta velocità di trasmissione interna dei codici e quindi da più rapida esecuzione dei programmi.

MEMORIE RAM E ROM

La memoria RAM (Random Access Memory = Memoria ad accesso casuale) è quella sulla quale si possono sia scrivere che leggere dei dati, i quali si possono modificare od anche annullare. La memoria di lavoro di un elaboratore, quella indicata sulle caratteristiche elettriche, è del tipo RAM. Quando la macchina viene spenta, si perdono tutte le informazioni memorizzate. Per questo motivo, occorre preventivamente "salvare" dati e programmi su cassetta o floppy disk.

La memoria ROM, (Read Only Memory = Memoria a sola lettura) contiene istruzioni fisse e non modificabili dall'esterno. In ogni calcolatore esiste una memoria ROM che contiene tutti i dati e le istruzioni relativi al Sistema

Operativo interno e alle istruzioni fisse come quelle che riguardano le operazioni matematiche, ecc.

SCELTA DEL CALCOLATORE

Le caratteristiche salienti di un calcolatore, a parte il SOFTWARE (programmi) che ha a disposizione, sono la memoria RAM di lavoro e la memoria di massa (CASSETTE, FLOPPY o HARD DISK). Un altro dato di cui bisogna tener conto è quello relativo alla possibilità di espansione sia per la RAM che per la memoria di massa.

La scelta dell'elaboratore da impiantare dipende dal tipo di lavoro che si deve svolgere. Se i programmi e i dati da inserire in RAM sono molto lunghi e complessi, sarà predominante il valore in Kbytes della memoria RAM di lavoro.

Se, per contro, il programma non è eccessivamente lungo ma i dati da memorizzare sono in numero elevatissimo (come ad esempio nella gestione di un indirizzario), è bene che sia la memoria di massa ad essere privilegiata. Per piccole aziende è sufficiente un calcolatore di 64 - 128 kbytes RAM e memoria di massa di 1 Megabytes (1 milione di bytes per dischetto). In questo caso, il dischetto viene registrato da ambo le parti con solchi magnetici molto più fini del normale. (FLOPPY da 5", in realtà sono 5" e 1/4 -DOPPIA FACCIA - ALTA DENSITA').

Per fornire un'idea più precisa del contenuto di un dischetto, si tenga conto che ad un byte corrisponde un carattere e che 1 milione di caratteri corrisponde all'incirca a 1000 pagine dattiloscritte.



Esercizio n. 2
RUN.

Lezione pratica

Vedere il RUN dell'esercizio n. 2.

L'esercizio n. 2 costituisce un miglioramento del numero 1, in quanto aiuta l'operatore ad apportare modifiche nelle spaziature e nelle tabulazioni (che per il momento sono automatiche cioè fornite con delle virgole poste dopo la istruzione PRINT).

È necessario che fin dai primi esercizi ci si abitui all'armonia compositiva. L'elaborato (PRINTATO) deve risultare ben distribuito verso il centro dello schermo e il tutto deve apparire armonioso alla vista.

NEW

Quando si ha in macchina un programma e si intende impostarne uno nuovo, va scritta l'istruzione NEW (nuovo). Tutto quanto eseguito in precedenza scompare dalla memoria RAM dell'elaboratore, anche se momentaneamente rimane stampato sullo schermo video. Premendo i tasti SHIFT/CLR, anche lo schermo verrà ripulito ed il cursore ritornerà in alto a sinistra per riprendere di nuovo il lavoro.

PRINT: PRINT: PRINT ecc.

Volendo spaziare più interlinee (in questo caso 3), non è necessario sprecare 3 righe di programma. Basterà scrivere i PRINT sulla stessa riga uno di seguito all'altro, purchè siano intervallati da due punti(:) che fanno da separazione fra le singole istruzioni.



A sinistra, procedimento per il salvataggio del programma su cassetta. L'istruzione appropriata è SAVE "SIGLA PROGRAMMA". Più in basso, quella per la verifica del salvataggio. Dopo aver riavvolto il nastro fino al numero di giri iniziale, si scrive l'istruzione VERIFY "SIGLA PROGRAMMA".

SAVE

Prima di spegnere il calcolatore, qualora interessi salvare il listato del programma, si scriverà l'istruzione **SAVE "SIGLA"**, mettendo fra gli apici la sigla o il nome del programma stesso. Sullo schermo comparirà la scritta **PRESS RECORD AND PLAY**, cioè premi sul registratore contemporaneamente i tasti **Record** (registrazione) e **Play** (avvio) (vedi figura). Al termine dell'operazione lo schermo fornirà l'informazione **SAVING (SIGLA PROGRAMMA) - OK - READY**. A questo punto si può spegnere la macchina. Il programma rimane custodito nella cassetta e potrà in seguito essere recuperato.

NOTA BENE

È importante durante questi primi esercizi la ripetizione dei piccoli programmi per prendere dimestichezza con la tastiera e acquisire velocità di esecuzione. Ha pure notevole rilievo l'impostazione esatta delle mani. La mano sinistra stazionerà in prevalenza vicino allo **SHIFT** di sinistra, quella destra con l'anulare e il mignolo in prevalenza sui tasti **CRSR →** e **CRSR ↑**. Risulterà più facile l'effettuazione delle correzioni. Naturalmente, entrambe le mani possono essere impiegate entrambe per scrivere i testi. Evitare di battere **SHIFT** e altri tasti contemporaneamente con la stessa mano.

BASIC SCHOOL - Lezione n. 3

Notizie di informatica

INTERFACCIA

Le diverse periferiche che vengono collegate all'elaboratore centrale quasi mai hanno un funzionamento omogeneo con quest'ultimo. Tanto per fare un esempio, la tastiera alfanumerica è costituita da pulsanti che, quando siano premuti, chiudono un contatto.

Per poter dialogare con il microprocessore, è necessario che per ogni contatto chiuso, venga prodotto un codice di 8 bit (byte) da trasferire all'interno della macchina. Si incarica di questa trasformazione una scheda speciale che si chiama INTERFACCIA (Fig. 3). Ogni apparecchiatura applicata al computer come la stampante, il lettore di floppy disk oppure il registratore a cassetta, il monitor ed altre eventuali, sono collegate al corpo macchina per mezzo di una scheda interfaccia.

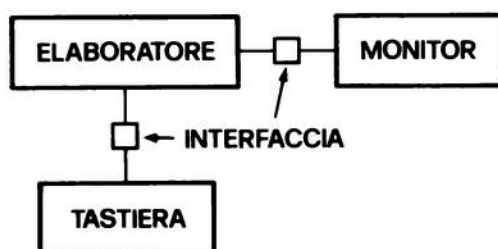


Fig. 3

TRASMISSIONI DATI

Per trasmettere dati (leggi bytes) da un apparato all'altro (es.: dal computer alla stampante oppure dal computer al registratore a cassette), vi sono 2 sistemi, quello parallelo e quello seriale.

TRASMISSIONE PARALLELA

Si avvale di un connettore a 8 fili (+ la massa) e trasmette 8 bit (1 byte) per volta, in contemporanea. Si tratta di un tipo di trasmissione molto veloce.

TRASMISSIONE SERIALE

Si avvale di un connettore semplice con 1 solo collegamento di uscita (+ la massa) e trasmette 1 bit dopo l'altro, come avviene per il registratore a cassette.

È una trasmissione molto lenta, giustificata soltanto dalla minor spesa sopportata, oppure dalla necessità di servirsi di cavi telefonici come nel caso di trasmissione di dati a distanza. Il numero di bit/sec trasmessi si chiama Baud.

MODEM

Nel caso di trasmissione seriale, all'ingresso della linea telefonica di chi invia i dati e all'uscita di quella di chi li riceve è necessario inserire un apparecchio di trasformazione da digitale a modulato (analogico) e viceversa chiamato MODEM (MODulatore-DEModulatore). Il senso di trasmissione è bidirezionale, ma alternativo.

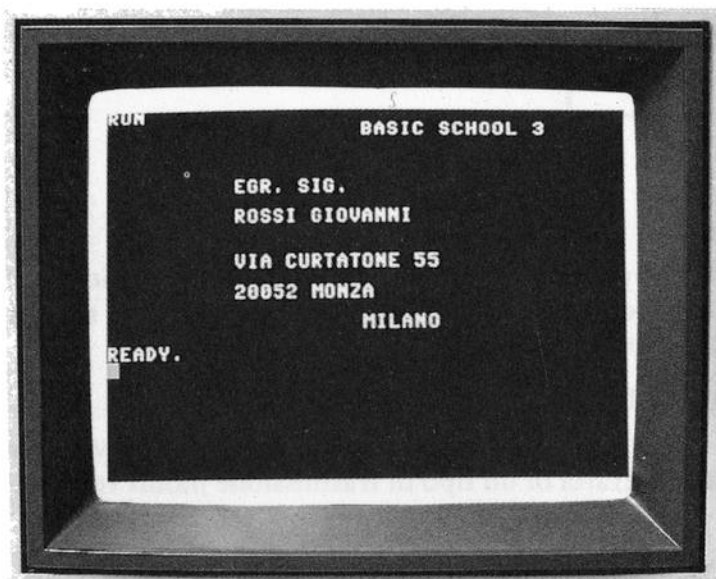
OFFICE AUTOMATION

È quella parte dell'informatica, attualmente in tumultuosa evoluzione che si interessa della introduzione dei computer nel lavoro d'ufficio. Il programma di lavoro più utilizzato sui computer d'ufficio è il WORD PROCESSING. Tramite esso, si può gestire elettronicamente una grande quantità di testi con illimitate possibilità di correzioni.

Si possono altresì raccogliere grandi quantità di indirizzi, spedire circolari e lettere con indirizzi diversi, tenere archivi, ecc.

Lezione pratica

Riprendere l'esercizio n. 2 e modificarlo come indicato nel RUN n. 3. Altre varianti si possono introdurre per completare l'apprendimento della scrittura tabulata in senso orizzontale e ben spaziata in senso verticale.



**Esercizio n. 3
RUN.**

Ripetere questi primi esercizi più volte, eventualmente introducendo piccole modifiche e controllando a ripetizione il LIST e il RUN. Ciò con l'evidente scopo di acquisire velocità e precisione e per rendersi meglio conto della interdipendenza fra le linee del programma e il risultato ottenuto sul video.

TAB()

Occorre imparare a tabulare con un numero di battute variabili e non soltanto col tabulato fisso. Es.:

10 PRINT TAB(25) "BASIC SCHOOL 3"

Ricordarsi che l'espressione TAB() va scritta tutta attaccata e che fra parentesi va posto il numero di battute di cui si intende tabulare. Es.:

30 PRINT TAB (10) "ROSSI GIOVANNI"

Qualora si intenda tabulare in più o in meno, basterà spostare il cursore con **CRSR→** e **CRSR←** sul numero compreso entro le parentesi e cambiare in più o in meno questo numero.

L'impiego del TAB dà all'operatore una maggior libertà nella composizione armoniosa degli elaborati (dicansi anche PRINTATI).

SHIFT+INST

In sede di correzione di una riga di programma già immagazzinata con il RETURN, può essere necessario creare degli spazi per aggiungere lettere o frasi. In questo caso, si porta con **CRSR** il cursore all'inizio degli spazi che si vogliono creare e si premono contemporaneamente i tasti **SHIFT** e **INST**. È utile calcolare prima il numero di spazi da produrre e una volta che questi si sono ottenuti con esattezza, si scriverà la parte di testo aggiuntiva, quindi si batterà il tasto **RETURN**. È bene, almeno per questi primi tempi, attenersi scrupolosamente a questa norma, perchè con lo **SHIFT/INST** è facile impantanarsi in pasticci grafici da cui un principiante non riesce ad uscire tanto facilmente.

Notizie di informatica

HARDWARE

Si tratta di tutto ciò che concerne il computer sotto l'aspetto costruttivo e funzionale (circuiteria, parte meccanica, servizio di assistenza).

SOFTWARE

Riguarda la parte soffice del computer, cioè i programmi.

BASIC

Linguaggio per principianti per mezzo del quale è possibile dialogare con il calcolatore per fargli eseguire dei programmi più o meno semplici.

È il linguaggio più vicino all'operatore in quanto egli si rivolge alla macchina con istruzioni espresse in lingua corrente (inglese). Ha il difetto di essere lento in fase esecutiva dovendo fare molti passaggi affinché un ordine in inglese venga trasformato in linguaggio macchina (in bytes), l'unico che il calcolatore capisce.

ASSEMBLER

È un linguaggio a livello macchina, per persone esperte. Ogni microprocessore ha un suo tipo di assembler. Consta di sigle operative come MOVE-ADD-LOAD con le quali è possibile scindere i programmi in istruzioni elementari che rappresentano i diversi passaggi dei bytes da un circuito all'altro, all'interno della macchina. L'uso di un cosiddetto ASSEMBLATORE serve a tradurre quelle sigle direttamente in bytes (10001111, ecc.) le sole che i circuiti elettronici digitali riescono a capire.

CIRCUITO INTEGRATO

Trattasi di una piastrina di silicio di pochi mm di lato, la quale è capace di contenere (integrati) anche molte migliaia di altri circuiti elementari della tecnica binaria (porte logiche, flip-flop) a loro volta costituiti da molti microtransistor, diodi, ecc. La parte essenziale del circuito integrato è chiamata anche CHIP ed è contenuta in un agglomerato scuro di materia plastica o ceramica dal quale escono dai 2 lati longitudinali i piedini per l'assemblaggio.

CHIP

La piastrina del circuito integrato che contiene i microcircuiti a transistor in grandissima quantità. È collegato ai piedini di uscita con dei sottilissimi fili d'oro.

ASCII

Codice internazionale che assegna ad ogni lettera o numero o altro carattere della tastiera del computer un codice di 8 bit (1 byte) ben definito.



Procedura per il caricamento del programma da cassetta a memoria RAM del calcolatore. L'espressione valida è LOAD "SIGLA PROGRAMMA". Scrivendo semplicemente LOAD, il computer carica il primo programma che trova registrato. SEARCHING FOR significa: Sto cercando. FOUND: Trovato. LOADING: Caricamento. READY: Caricato.



Quando si carica un programma di cui non si conosce l'esatta ubicazione sulla cassetta, compaiono sullo schermo le sigle dei programmi via via trovati lungo il percorso. Ad ogni FOUND il registratore si arresta per qualche secondo. Per farlo ripartire immediatamente, premere il tasto **⏪**.

Esercizio pratico

LOAD

È l'istruzione che occorre dare al calcolatore per caricare il programma in memoria dalla cassetta dove era stato salvato in precedenza con l'istruzione SAVE.

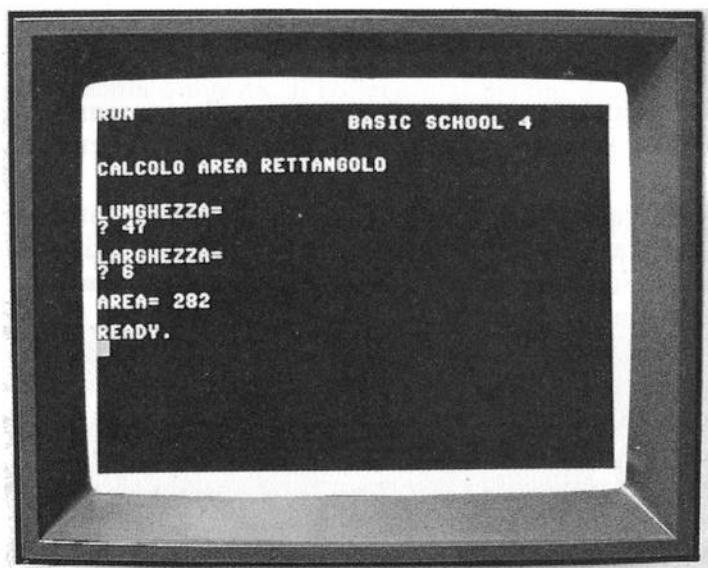
Es.: LOAD "3" significa "carica il programma avente la sigla 3" oppure LOAD "ARTURO" significa "carica il programma salvato con la sigla ARTURO". Attenzione che la sigla del LOAD sia **identica**, sia come frase che come spazi a quella utilizzata per il SAVE.

Dopo aver scritto la formula di caricamento, si preme il tasto RETURN e sullo schermo compare la scritta: PRESS PLAY ON TAPE (premi il tasto PLAY sul registratore). Naturalmente prima dell'operazione di LOAD, occorre riavvolgere il nastro per ritornare all'inizio (meglio un paio di giri in anticipo) del programma da caricare. Per fare questo ci si avvale del contatore di giri posto sul registratore. I numeri di inizio e di fine del programma in oggetto, saranno infatti stati scritti in fase di salvataggio su una apposita etichetta applicata al contenitore della cassetta.

N.B. Durante le operazioni di caricamento (LOAD), ad un certo punto, il registratore si ferma e sullo schermo compare la frase FOUND "SIGLA PROGRAMMA", cioè è stato trovato il programma in oggetto. Se questa sigla corrisponde a quella indicata, dopo qualche secondo viene ripreso il recupero del listing. Per accelerare questa operazione, si può premere il tasto Commodore **⌘** (l'ultimo in basso a sinistra).



Esercizio n. 4
RUN parziale.
Il calcolatore è fermo
al primo INPUT
in attesa
che gli venga fornito
il primo dato
dall'esterno.



**Esercizio n. 4
RUN completo.**

INPUT

Si tratta ora di eseguire il programma che compare nei RUN n. 4 parziale e RUN n. 4 completo.

In questo esercizio viene mostrata per la prima volta l'istruzione INPUT. L'allievo scriverà sullo schermo il testo del programma fino ad arrivare alla frase LUNGHEZZA= sulla falsariga di quanto fatto negli esercizi precedenti. A questo punto occorre introdurre il concetto di INPUT e di "dato variabile" o VARIABILE.

I primi 3 esercizi hanno mostrato un uso incompleto del computer. Infatti fin qui esso è servito soltanto per scrivere sul monitor delle semplici frasi. Ma nell'applicazione reale, il suo uso è volto principalmente alla gestione di dati (numerici o letterali), dati che possono venir utilizzati a piacimento per le elaborazioni che si rendono necessarie (si pensi ad es. alla gestione di un magazzino, in cui i numeri relativi ai pezzi venduti cambiano in continuazione).

La parola INPUT significa "metti dentro". Quando l'elaboratore arriva a leggere la riga di programma nella quale è inserito un INPUT, fa all'operatore esterno un punto di domanda (?) alla quale occorre dare una risposta, altrimenti la macchina non continua l'esecuzione del programma. A questo punto di domanda occorre rispondere con un dato numerico se la sigla della variabile è costituita di una o due lettere. Es.:

```
200 INPUT LU
```

(LU è la sigla che diamo alla LUNghezza, ma potrebbe essere anche P o Q o S oppure L se non ci fosse il rischio di confonderla con la Larghezza).

Se, invece, la sigla che diamo alla variabile è una lettera seguita dal segno \$ (dollaro), dobbiamo fornire in risposta una cosiddetta "STRINGA" cioè, o un dato completamente letterario oppure un dato misto di lettere e numeri. Es.:

```
200 INPUT P$
```

(P\$ potrebbe essere un nome, o un cognome, oppure una via ecc.).

N.B. Occorre sempre accompagnare all'INPUT la sigla della variabile altrimenti il calcolatore non saprà identificarla quando dovrà servirsi dei dati per le elaborazioni richieste. Es.:

```
100 INPUT LU
110 INPUT LA
120 A = LU x LA
```

dove A (area) è una variabile interna il cui valore dipende dai numeri attribuiti a LU (lunghezza) e LA (larghezza).

Ricordare che la moltiplicazione va effettuata con il tasto * e non con la x (ics).

REM

Significa REMark (Nota bene) e serve per scrivere una frase esplicativa che però non viene letta come riga di programma, ma resta soltanto nel listing. In particolare quando si abbiano programmi molto lunghi, il REM serve all'operatore per meglio identificare le diverse sezioni di cui esso si compone, sia per lo studio che per apportare delle modifiche.

RUN/STOP

Questo tasto va premuto quando, per qualsiasi ragione, si voglia interrompere un programma in esecuzione oppure lo svolgimento di un listing.

Se assieme a RUN STOP si preme il tasto **RESTORE**, tutte le variabili vengono azzerate e si ritorna con il READY all'inizio superiore dello schermo.

CONTROL

Questo tasto serve per far procedere un listing oppure un run lentamente, riga dopo riga.

Notizie di informatica

STAMPANTE

È l'apparecchiatura che ha il compito di stampare su carta quella parte del programma che è destinata all'archivio. Lo stesso "listato" del programma può essere trascritto con la stampante. L'elemento grafico può essere una matrice ad aghi (es.: 9x7 aghi).

Ad ogni segno grafico, vengono proiettati in avanti quegli aghi che sono interessati per lasciare traccia del segno stesso. Gli aghi sbattono sulla carta tramite un nastro inchiostro.

Vi sono anche stampanti di tipo diverso, come quelle a margherita. In genere, queste ultime, sono delle macchine per scrivere elettroniche, che possono venire comandate, attraverso una scheda interfaccia, da un computer. Sono molto più lente (circa 17 caratteri/sec) di quelle ad aghi che possono arrivare anche a stampare 240 ed oltre caratteri al secondo (stampando sia nel senso di andata che in quello di ritorno). Vi sono poi stampanti più moderne a getto d'inchiostro, ancora in via di perfezionamento, in ogni caso molto costose.

GRAFICA E SEMIGRAFICA - PLOTTER

Gli aghi della matrice della stampante possono venire programmati per fare curve e disegni (semigrafica). Per la grafica vera, occorre invece avvalersi di stampanti apposite con pennini (eventualmente impieganti inchiostri diversamente colorati) che possono tracciare qualsiasi tipo di disegno (es.: scocche di automobili, disegni di meccanica, ecc.). Queste particolari stampanti vengono chiamate PLOTTER.

IEEE

Sigla dell'Associazione degli Ingegneri Elettronici Americani. Ha la corrispondente in Italia nell'AEI (Associazione Elettronici ed Elettrotecnici Italiani). Si interessa di normativa tecnica e standardizzazione delle prese e spine di accesso ai computer. Es.: presa IEEE 488.

LOGICO, BINARIO, DIGITALE

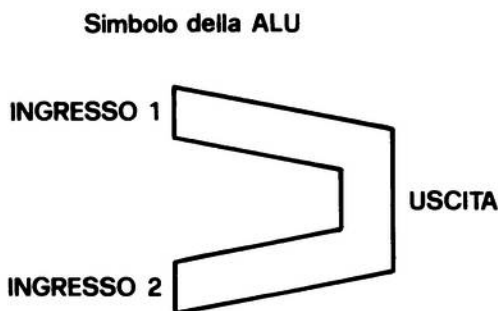
Tre espressioni che, con diversi significati specifici, fanno parte della tecnica dei computer. Questa tecnica, come si sa, si avvale di soli due stati elettrici di base: Presenza di tensione (simbolo 1), assenza di tensione (simbolo 0).

Più precisamente, la parola "logico" si riferisce all'impiego di circuiti cosid-

detti logici come le porte, i flip-flop, ecc.; “binario” riguarda il calcolo aritmetico in base 2 (aritmetica di Boole) che fa da base al calcolo degli elaboratori; “digitale” che significa “cifrato” si riferisce a tutti quei circuiti che sono capaci di contare ed eventualmente di estrarre i prodotti del calcolo con indicatori luminosi tipo “display” e simili.

ALU

Significa Arithmetic Logic Unit (Unità aritmetica-logica). (Fig. 4).



È una sezione interna al microprocessore nella quale vengono svolti i calcoli e i confronti logici. Es.:

```
20 A = 30
30 B = 40
40 S = 30+40
```

oppure

```
20 INPUT A
30 INPUT B
40 P = A x B
50 IF P = 300 THEN PRINT "OK"
```

Nella riga 50 viene prospettato un confronto logico, poichè se il codice binario del valore numerico del prodotto P risulta uguale al codice binario del numero 300, allora si attiva il comando che scrive sullo schermo la parola “OK”.

Lezione pratica

Osservare il RUN (esecuzione) dell'esercizio n. 5 parziale e completo e scriverne la lista di programma. L'esercizio n. 5 è sensibilmente uguale al n. 4 eseguito in precedenza con una sola significativa eccezione, le istruzioni PRINT e INPUT sono state incorporate in una sola riga.

Il vantaggio di questa nuova procedura è di avere, in fase esecutiva, il punto

di domanda allineato alla frase di richiesta. Migliora anche il rendimento estetico dell'elaborato che compare sul video.

```
20 INPUT "LUNGHEZZA =" ; LU
```

In genere, quando si programma un INPUT (dato variabile fornito dall'esterno in sede esecutiva), occorre anche dare sullo schermo spiegazione del dato richiesto, altrimenti il calcolatore offre soltanto un punto di domanda (?) senz'altra dicitura e per chi non abbia assistito alla preparazione del programma stesso, non è facile intuire che cosa in verità venga richiesto.



Esercizio n. 5
RUN parziale.
Si attende il dato
della lunghezza.



Esercizio n. 5
RUN completo.

Per questo motivo, nell'esercizio precedente si scrissero su una riga il PRINT esplicativo e nella riga seguente l'INPUT. Con questa nuova procedura, prima si scrive l'istruzione INPUT poi si inserisce fra i due apici la "FRASE" come se si trattasse di un PRINT, quindi **si fa seguire la VARIABILE intervallando con un punto e virgola (;)** Es.:

```
20 INPUT "LARGHEZZA =" ; LA
```

La riga di programma va così spiegata: scrivi sullo schermo la frase "LARGHEZZA=" al termine della quale fai un punto di domanda (INPUT) attribuendo alla VARIABILE la sigla LA.

N.B. Nel caso presente, se avessimo chiamato la Lunghezza con la variabile L, chiamando anche la Larghezza con L, faremmo confondere il calcolatore che non saprebbe più separarle. Esso riconosce per i dati numerici al massimo le prime due lettere. Variabili con una sigla più lunga non avrebbero alcun senso.

Attenzione

La frase introdotta con INPUT "FRASE", non si può tabulare come un PRINT. Si vedrà più avanti come è possibile spostarla a piacimento sullo schermo.

Il punto e virgola (;) va posto prima di una variabile soltanto in questo caso specifico in cui all'input si associa una frase da printare.

BASIC SCHOOL - Lezione n. 6

Riassunto delle espressioni più significative delle precedenti lezioni:

PRINT

| | |
|---------------------------------|---|
| PRINT | Spaziare in verticale di una interlinea. |
| PRINT:PRINT:PRINT , ecc. | Spaziare più interlinee in verticale. |
| PRINT "FRASE" | Scrivi sullo schermo la frase fra gli apici. |
| PRINT,, "FRASE" | Scrivi sullo schermo la frase fra gli apici calcolando 2 tabulazioni automatiche (ogni virgola sposta il cursore a destra di 10 battute partendo dal margine sinistro). |
| PRINT TAB(15) "FRASE" | Scrivi sullo schermo la FRASE fra gli apici, tabulando 15 battute verso destra. |
| PRINT A | Scrivi il valore in precedenza attribuito oppure calcolato per la variabile A. |
| PRINT "AREA=" A | Scrivi la frase "AREA=" e di seguito il valore calcolato o attribuito alla variabile A. |

N.B. Nella compilazione del programma, invece di scrivere la istruzione **PRINT**, è possibile semplificare battendo il punto interrogativo (?). In questo libro di base, tuttavia, si è preferito scrivere sempre **PRINT** per esteso per non ingenerare confusione all'allievo durante la lettura dei listati che, in tal modo, risultano più immediatamente comprensibili.

La stessa cosa vale anche per tutte le altre numerose abbreviazioni che sono possibili col Commodore, abbreviazioni che eventualmente riprenderemo nel futuro corso di Basic superiore attualmente in preparazione presso la Basic School.

DEL

INST + SHIFT

Quando in fase di correzione di una riga di programma già messa in memoria (col **RETURN**) si intende spostare a sinistra una parte della riga per eliminare uno spazio oppure a destra per crearlo, si porta il cursore sul primo segno della sezione mobile e si preme:

DEL (Delete) per chiudere lo spazio (spostamento a sinistra)

INST + SHIFT per aprire lo spazio (spostamento a destra). **INST=Insert**.

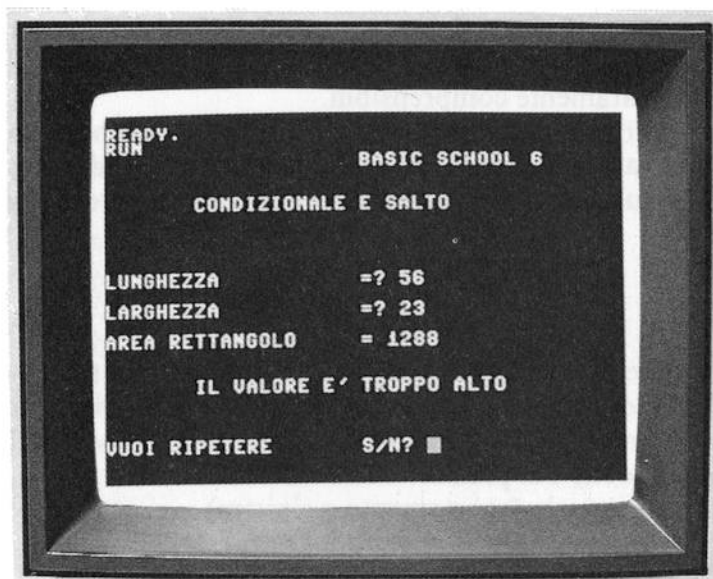
A correzione eseguita, battere il **RETURN**.

Esercizio pratico

Scrivere il programma relativo al RUN dell'esercizio n. 6 (parziale e completo) mostrato nelle fotografie.



**Esercizio n. 6
RUN parziale.
Siamo al 1° INPUT.**



**Esercizio n. 6
RUN completo.
Il calcolatore
è all'ultimo INPUT
per scegliere
se chiudere o ripetere
il programma.**

Le istruzioni nuove introdotte con questo esercizio sono:

IF...THEN

(Significa: SE...ALLORA). Con questa istruzione condizionale, il computer viene abilitato a prendere delle decisioni in base ad una condizione imposta dall'operatore e che si deve verificare per darle corso. Es.:

```
160 IF R$="S" THEN PRINT "☐" : GOTO 10
```

Significa: Se la variabile R\$ assume la forma della lettera S, allora pulisci lo schermo e riparti dalla riga di programma n. 10 (cioè ripeti il programma stesso. (PRINT "☐" è il segno grafico di PRINT "SHIFT+CLR").

Quando l'istruzione IF...THEN viene impiegata insieme all'altra GOTO (VAI A) ne nasce una funzione che prende il nome di SALTO CONDIZIONATO.

GOTO

(Significa VAI A). Obbliga il computer a fare un salto nel programma, nel senso di riprendere da una riga ben definita che può essere sia inferiore come numero (salto all'indietro o ripetizione) oppure superiore (salto in avanti o scavalco di una parte del programma). Ancora,

```
170 IF R$="N" THEN END
```

Significa:

Se la variabile stringa (non numerica pura) R\$ è = alla lettera N, allora chiudi il programma.

\$(dollaro)

Segno che va posto di seguito ad una o due lettere, per indicare una cosiddetta VARIABILE STRINGA. Si tratta di un INPUT (punto di domanda) cui verrà fornito un dato letterario (lettera o parola) oppure misto.

```
60 Azzeramento delle variabili
```

Quando un programma di calcolo viene ripetuto senza passare dal RUN occorre che prima di fornire i nuovi dati (Input) vengano azzerate le variabili relative a questi input. In caso contrario, qualora in fase di input al posto del numero fosse battuto il solo Return, il calcolo finale risulterebbe alterato per il fatto che il calcolatore presenterebbe i risultati ottenuti in precedenza. Ciò non avviene quando si ricomincia il programma con il RUN, perchè in questo caso la variabili vengono automaticamente azzerate.

Lezione pratica

Eeguire gli esercizi n. 7 e 8. Si tratta di elaborazioni e perfezionamenti dell'esercizio n. 6. Lo scopo che ci prefiggiamo con questi due esercizi è quello di migliorare l'armonia compositiva dei printati. Vengono inoltre introdotte alcune nuove istruzioni.

PRINT "♥" (esercizio n. 7)

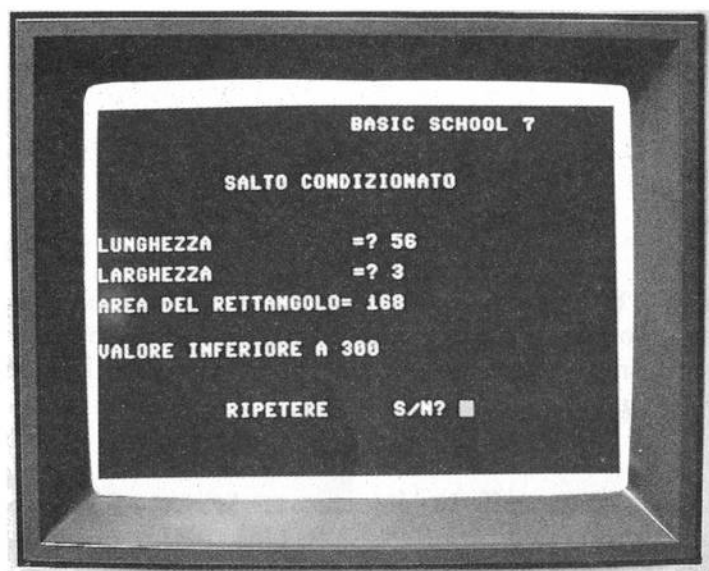
Con questa istruzione, il calcolatore effettua interamente la stessa funzione ottenibile premendo dall'esterno i tasti SHIFT+CLR, cioè ripulire lo schermo e riportare il cursore in alto a sinistra. Va posto all'inizio di ogni programma. Per ottenere il segno grafico dopo il PRINT, premere l'apice e quindi i tasti SHIFT e CLR contemporaneamente.

INPUT "□□□□□□ RIPETERE S/N" (esercizio n. 7)

Come si sa, l'INPUT non accetta tabulazioni né automatiche (con virgola) né col TAB. Per spostare verso destra l'inizio di una frase introdotta con l'INPUT, vi sono due sistemi:



Esercizio n. 7
RUN completo.
L'area ottenuta
è uguale a 300
cioè OK. Siamo
in attesa
della decisione finale
(l'ultimo INPUT).



Esercizio n. 7
RUN completo.
L'area è inferiore a 300.
Siamo in attesa
della decisione
finale (ultimo INPUT).

— Dopo il primo apice battere la barra spaziatrice di tante battute quante ne necessitano di tabulazione.

— Dopo il primo apice, battere con il tasto **CURSORE A DESTRA (CRSR→)** le battute di cui si vuole tabulare la frase indicata fra virgolette.

Questo secondo sistema, adottato nella riga di programma n. 190 dell'esercizio n. 7 è il migliore, perché non cancella eventuali scritte printate in precedenza sullo schermo.

```
IF...THEN GOTO 10
IF...THEN 10 (esercizio n. 7)
```

Quando il condizionale porta il programma a saltare su una determinata riga, si può omettere il **GOTO**.

R\$

(leggi R dollaro oppure R stringa, esercizio n. 7)

Il segno \$ accoppiato ad una variabile la trasforma da numerica a "stringa" cioè letterale o mista di lettere e numeri. Questa variabile viene adoperata

come sigla per INPUT ai quali si debba rispondere con lettere oppure con numeri e lettere insieme.

IF R\$="S" (esercizio n. 7)

Significa: se la variabile stringa R\$ è uguale a "S", cioè se all'INPUT rispondo con una S. Ricordare che quando la variabile è stringa, la stringa va posta entro gli apici o virgolette, a differenza di quanto avviene per le variabili numeriche pure.

Esempi:

IF R = 2

IF R\$ <> "S" (esercizio n. 7)

Se la variabile stringa R\$ è diversa <> dalla lettera S, cioè se ripondo con un altro qualsiasi carattere, il programma si chiude.

PRINT "  AREA RETTANGOLO =" (esercizio n. 8)

Il segno grafico si ottiene premendo subito dopo il primo apice, il tasto `CRSR→` da solo, cioè il `CURSORE A DESTRA`. Nel PRINT, si può tabulare in altri modi, ma per piccole tabulazioni, questo sistema può essere preferito.



Esercizio n. 8
RUN parziale,
fermo al 2° INPUT.

GET (esercizio n. 8)

Serve ad effettuare un comando direttamente dalla tastiera senza passare per il RETURN. La formula è la seguente:

- 260 GET R\$ (esplora le stringhe della tastiera)
- 270 Quando venga battuta la stringa chiamata S, vai alla riga n. 10 (cioè ripeti il programma)
- 280 Quando venga battuta la stringa chiamata N, chiudi il programma
- 290 Ritorna ad esplorare la tastiera (cioè ritorna alla riga di programma del GET, la n. 260).



Esercizio n. 8
RUN completo. Siamo in attesa della decisione finale, ottenuta con la funzione GET, cioè direttamente da tastiera. Come si può notare, col GET manca il cursore lampeggiante che segnala l'attesa della risposta S oppure N. Nei programmi migliori, il cursore pulsante viene creato artificialmente con un'apposita subroutine.

BASIC SCHOOL - Lezione n. 8



Esercizio n. 9
RUN completo.

Lezione pratica

FOR...NEXT

Inserire nel programma l'esercizio n. 9 che serve a introdurre l'operazione ripetitiva (ciclica) FOR...NEXT. ➤

```
FOR K=1 TO 10...NEXT K
```

Con questa espressione, il programma interno a FOR...NEXT viene ripetuto ciclicamente per 10 volte (al posto di 10 potrebbe esservi qualsiasi altro numero). (Fig. 5). Ad ogni ciclo, la variabile K assume il valore progressivamente crescente fino a quando, arrivata a 10, il ciclo si ferma.

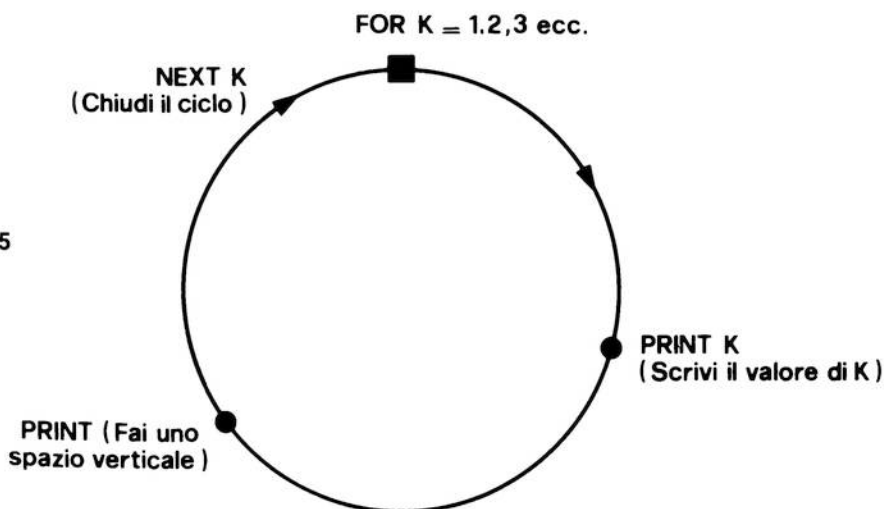


Fig. 5

Ricordiamo che ogni ciclo si chiude con la parola NEXT. Le istruzioni che sono fornite fra FOR e NEXT, vengono ripetute a ogni ciclo. Es.: se fra FOR e NEXT vengono date in successione le istruzioni PRINT K e PRINT, ad ogni passaggio viene scritto il valore della variabile K assunto in quel momento e creato uno spazio in verticale prima di scrivere il valore successivo.

UTILIZZAZIONE DEL FOR...NEXT

Spaziamento in verticale

Quando debbano crearsi delle spaziature verticali di notevoli dimensioni, al posto di scrivere tanti PRINT, si può usare l'espressione:

```
FOR K = 1 TO N : PRINT : NEXT K
```

dove N è il numero delle interlinee desiderato.

Linea di separazione

Volendo separare due parti di un elaborato con una linea bianca continua orizzontale, si può usare l'espressione:

```
FOR K = 1 TO 40 : PRINT "☐" NEXT K
```

dove "☐" è il segno grafico che si ottiene premendo (dopo aver scritto l'apice) lo SHIFT e la lettera, C. Diciamo per inciso che se si preme il tasto Commodore al posto di SHIFT, invece del segno ☐ viene stampato quello di sinistra ☐.

FOR...NEXT.....STEP

Si chiama "operazione ciclica a passo". In questo caso, la variabile non sale progressivamente, ma seguendo dei salti regolari. Es.:

```
FOR K = 1 TO 20 STEP 2
```

Ad ogni passaggio del ciclo, K varia di 2 unità. In questo caso specifico, K assumerà i valori di 1-3-5-7-9-11-13-15-17-19.

FOR...NEXT...STEP—

Si chiama operazione ciclica inversa, cioè il conteggio parte con la variabile avente il valore più alto per regredire fino ad assumere quello più basso. Es.:

```
FOR K = 20 TO 1 STEP —1
```

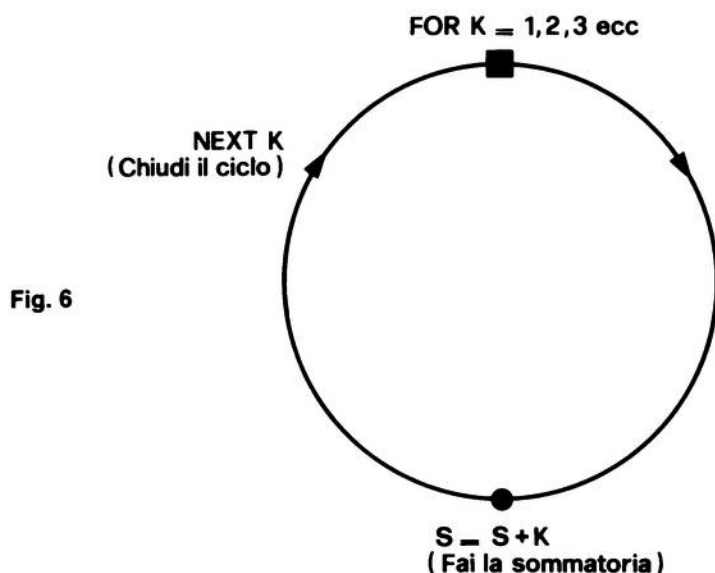
K assumerà progressivamente valori in diminuzione da 20 a 1.
Altro esempio:

```
FOR K = 20 TO 1 STEP —2
```

In questo caso si avrà una operazione ciclica inversa da 20 a 1. I valori di K saranno in successione: 20, 18, 16, 14, 12, 10, 8, 6, 4, 2.

Commento all'esercizio n. 9

Le righe 70-90 rappresentano una normale operazione ciclica di conteggio da 1 a 20 con stampa in orizzontale dei valori assunti via via dalla variabile K. Le righe 130-150, ripetono l'operazione ciclica precedente, ma all'interno del LOOP o CICLO, viene eseguita l'espressione $S=S+K$ che serve a fare la sommatoria di tutti i valori assunti da K nei 10 cicli (vedere la spiegazione nell'Appendice 2) (Fig. 6).



Ad ogni ciclo, la variabile S si incrementa del valore che K assume in quel ciclo stesso. Così facendo, S rappresenta sempre la somma di tutti i valori che K ha assunto in precedenza. Esempio:

Per K = 1 $S = 0 + 1 = 1$
Per K = 2 $S = 1 + 2 = 3$
Per K = 3 $S = 3 + 3 = 6$
Per K = 10 $S = 45 + 10 = 55$
ecc. ecc.

Notizie d'informatica

CLOCK

La parola CLOCK significa orologio, cadenza, e sta ad indicare il funzionamento cadenzato, a scatti, del calcolatore. Tutte le operazioni elementari che si svolgono in un computer, avvengono a scatti, con una frequenza di ripetizione che può andare da 1.000.000 di scatti al secondo (frequenza di 1MHz = 1 Mega Hertz) fino a 5.000.000 di scatti al secondo (frequenza di lavoro del microprocessore di 5 MHz). Da questa frequenza dipende la velocità di

esecuzione dei lavori da parte del computer, la quale a sua volta è condizionata dal tipo di transistor impiegati e dalla circuiteria utilizzata per comporre il macchinario elettronico interno.

MULTIPROGRAMMAZIONE

Non tutti i circuiti elettronici interni funzionano contemporaneamente. Alcuni devono ogni tanto fermarsi per attendere che altri abbiano concluso il proprio lavoro.

Si approfitta delle pause d'attesa che i circuiti presentano per inserire nel calcolatore dei dati provenienti da diversi terminali. Ciò avviene nei grossi centri di elaborazione come nelle banche oppure nelle grosse ditte dove ad una macchina centrale (MAIN FRAME) sono applicati diversi videotermini.

BUS

Nella trasmissione dei dati elettrici fra un circuito e l'altro del computer, gli 8 bit costituenti un byte, vengono portati in giro simultaneamente su un fascio omogeneo di 8 conduttori che viene chiamato BUS (contrazione della parola AUTOBUS). All'interno del calcolatore vi sono 3 tipi di BUS:

- a) il **BUS DEI DATI** che è costituito da 8 piste di rame appaiate (per un computer a 8 bit). Serve per trasportare i BYTES da una parte all'altra del circuito.
- b) il **BUS DEGLI INDIRIZZI** che è costituito da 16 conduttori appaiati. Serve per trasportare una parola in codice formata da 16 bits che ha il compito di indicare quale cella della memoria RAM deve essere aperta per introdurvi o per estrarvi un determinato byte. Con 16 bits si possono avere 2^{16} indirizzi, da 0 a 64.000 circa, ($2^{16}=65.536$) tante sono le celle di memoria contenute nella RAM.
- c) **BUS DEI CONTROLLI**. Contiene dalle 5 alle 7 piste e serve per portare in giro singoli bit che comandano l'apertura o la chiusura di singoli circuiti, ad esempio, nel caso si intenda salvare un programma, per dare il via al registratore oppure al meccanismo di lettura del FLOPPY DISK.

BASIC SCHOOL - Lezione n. 9

Lezione pratica

(Con riferimento all'esercizio n. 10. Vedere il RUN)

```
60 GOSUB 230
```

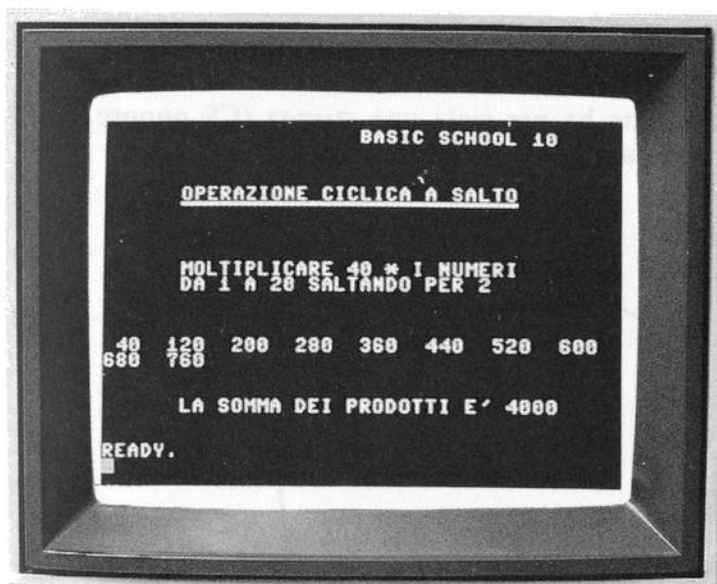
Rappresenta un salto in avanti nel programma per effettuare un sottoprogramma (chiamato in gergo **SUBROUTINE**) che viene ripetuto più volte nel corso della esecuzione.

GOSUB

Significa: VAI A ed è seguito dal numero della riga in cui ha inizio il sottoprogramma.

```
230 FOR K = 1 TO 5 : PRINT : NEXT K : RETURN
```

È l'esempio di un sottoprogramma che serve per spaziare 5 interlinee verso il basso. Per 5 volte il calcolatore effettua un PRINT (una battuta a vuoto in verticale), quindi chiude il ciclo e ritorna (RETURN) alla riga di programma che segue immediatamente quella del GOSUB.



Esercizio n. 10
RUN completo.

È interessante vedere come diverse istruzioni omogenee, possono venire scritte nella stessa riga di programma, purchè siano separate dai due punti (:). Nelle subroutine raggiunte con il GOSUB, non dimenticare mai di chiudere con l'istruzione RETURN (che non ha niente a che vedere con l'omonimo tasto).

END

Per far sì che il calcolatore non consideri 2 volte il sottoprogramma, prima quando riceve l'istruzione GOSUB e poi, in fondo, quando raggiunge il sottoprogramma stesso, prima che questo abbia inizio o comunque prima della serie delle subroutine, va chiuso il programma principale con l'istruzione END.

Ricordarsi le espressioni:

```
INPUT LU
INPUT "LUNGHEZZA =" ; LU
```

va utilizzata una variabile che ha per sigla una lettera (A) oppure due lettere (LU) quando si richiede un dato di tipo numerico puro.

Trattandosi qui di una lunghezza, potrà venire risposto con 425 oppure 35.4 (in americano, nelle cifre la virgola è sostituita da un punto).

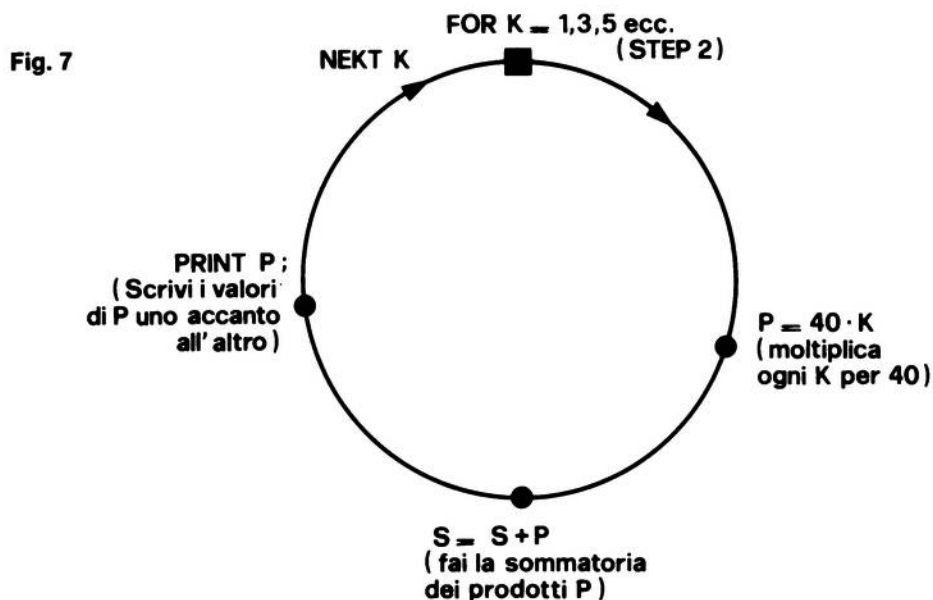
Soltanto quando il dato fornito debba essere per forza un numero intero, la variabile deve essere seguita dal segno %. Es.:

```
INPUT "ABITANTI" ; AB%
```

A questa domanda verrà certamente fornito un numero intero, in caso di errore il calcolatore non accetta il dato e indica che è stato commesso uno sbaglio.

```
INPUT C$
INPUT "COGNOME =" ; CO$
```

va utilizzata la variabile, che ha per sigla una stringa (C\$ oppure CO\$) quando si richiede in risposta un dato letterario o misto.



Trattandosi qui di un COGNOME, in risposta verrà fornito ad esempio ROSSI, ZAMPINI, ecc.

Commentando l'esercizio n. 10

Nella prima parte dell'esercizio, dopo l'enunciato delle righe 50 - 90 si crea una operazione ciclica (ripetitiva) per il conteggio dei numeri da 1 a 20 saltando di 2 in 2. (Fig. 7).

All'interno di ogni singolo ciclo viene calcolato il prodotto $P = 40 * K$, poi viene creata la sommatoria dei prodotti con la formula $S = S + P$ e quindi vengono scritti sullo schermo i valori di P uno a fianco all'altro (perché `PRINT P;` è seguito dal punto e virgola).

La stampa del valore di S è fuori dal ciclo `FOR...NEXT` e viene preso soltanto il valore finale (sommatoria dei prodotti).

Dopo la chiusura del programma (riga 180), viene sviluppata la subroutine per spaziare 3 righe in verticale. Essa si chiude con l'istruzione `RETURN` (ritorna alla riga che segue il `GOSUB 230`).

BASIC SCHOOL - Lezione n. 10

GOSUB

L'esercizio n. 11 (Vedere il RUN alla pagina che segue) completa le esperienze già avute sull'operazione ciclica FOR...NEXT. Sono da segnalare i seguenti passaggi:

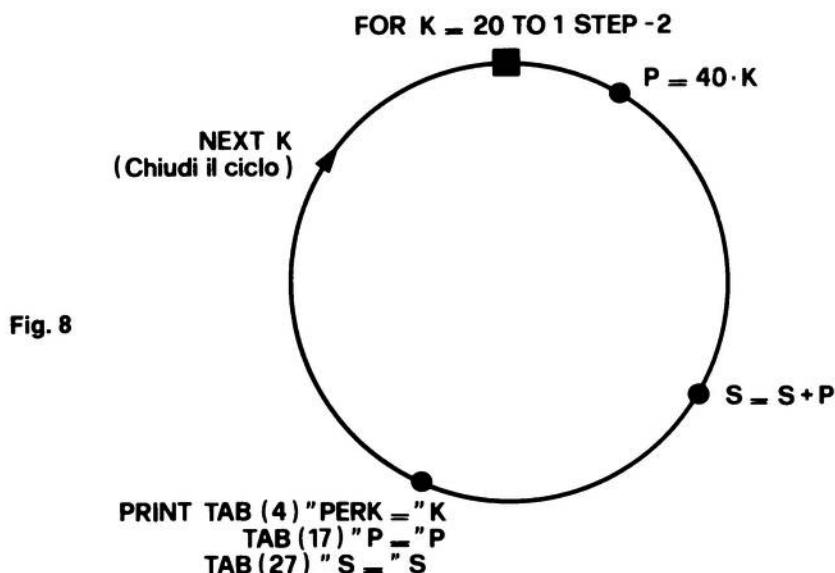


Fig. 8

20-50-80. Trattandosi di diversi spazi in senso verticale, viene utilizzata una unica subroutine posta sulla riga 150. Non sapendo a priori di quanto si intende spaziare e volendo utilizzare una sola SUB per tutti i casi di spaziatura verticale si introduce, a monte dell'istruzione GOSUB 150, la variabile SP (spaziare) alla quale verrà dato il valore che corrisponde alle righe di spazio desiderate.

```
20 SP=2 : GOSUB 150
150 FOR L = 1 TO SP : PRINT : NEXT L : RETURN
```

Naturalmente si tratta soltanto di un esempio didattico per introdurre il GOSUB. Gli spaziamenti col FOR...NEXT sono giustificati soltanto se in alternativa si devono scrivere molti PRINT

90 per K da 20 a 1, saltando per 2, quindi con K = 20,18,16, ecc. fino a 2
100 calcola ad ogni passaggio (o ciclo) il prodotto $P=40 * K$. In sequenza, si avranno i valori $P=40*20=800$, $P=40*18=720$ ecc.
110 calcola ad ogni passaggio la somma dei prodotti con la formula $S=S+P$
120 stampa tab. 4 la frase PER K= e scrivi il valore di K, poi stampa di



Esercizio n. 11
RUN completo.

seguito tab. 15 la frase P= e accanto il valore di P, ancora di seguito, stampa la frase S= e accanto il valore di S.

Le frasi avranno la seguente sequenza.

PER K= 20 P=800 S= 800

PER K= 18 P=720 S=1520

ecc. ecc.

Come si può vedere dallo sviluppo della riga di programma 120, è possibile scrivere sulla stessa linea una grande quantità di frasi e di variabili diversamente tabulate. Ciò è di grande utilità quando, ad esempio, si vogliono



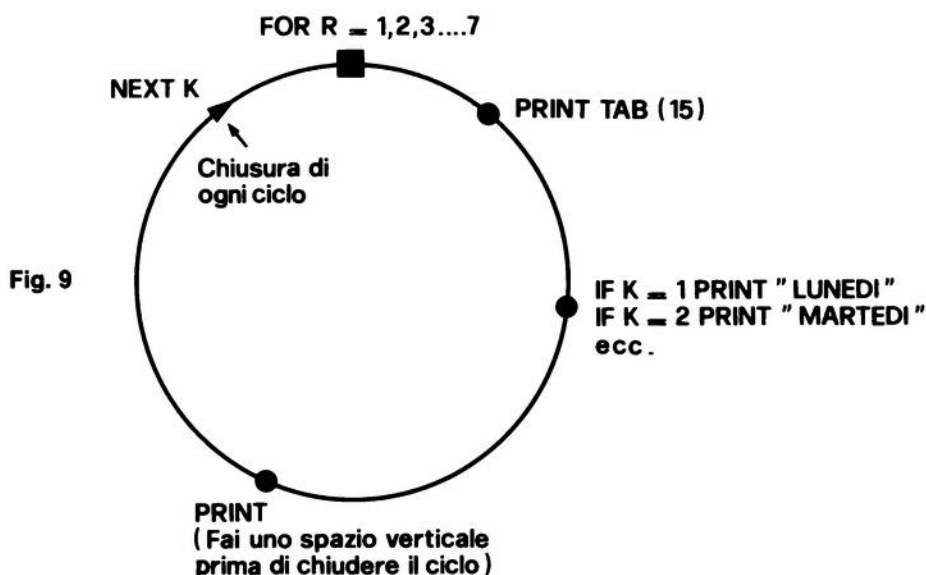
Esercizio n. 12
RUN completo.

compilare dei prospetti riassuntivi (es.: gestione dell'IVA), prospetti che è possibile anche stampare su carta.

140 Il programma viene chiuso affinché il calcolatore non legga come istruzione terminale la subroutine che interessa soltanto le righe 20-50-80.

Commento all'esercizio n. 12

Con questo esercizio (vedere il RUN relativo), si utilizza il FOR...NEXT non come ciclo di numeri ma come pura ripetizione. Esso infatti è legato alla istruzione IF...THEN e scrive frasi diverse a seconda del numero di ripetizione del ciclo. (Fig. 9).



```
80 Per 7 volte (K da 1 a 7)
85 Tabula 15 ad ogni ciclo
90 Se K = 1 scrivi LUNEDI'
100 Se K = 2 scrivi MARTEDI', ecc.
140 Fai uno spazio verticale prima della chiusura di ogni ciclo
170 Chiusura di ogni ciclo.
```

N.B. Si provi a ripetere l'esercizio 12 cambiando la riga 80 nella seguente:

```
80 FOR K=15 TO 21
```

si vedrà come sia possibile scrivere accanto a LUNEDI, MARTEDI, ecc., anche il numero del giorno corrispondente. Se poi, la riga 85 viene modificata in:

```
85 PRINT TAB(6) K "SETTEMBRE" SPC(5);  
90 IF K=15 THEN PRINT "LUNEDI"  
100 IF K=16 THEN PRINT "MARTEDI"  
ecc. ecc.
```

si otterrà sullo schermo un calendario settimanale più completo con le frasi disposte come segue.

```
15 SETTEMBRE  LUNEDI  
16 SETTEMBRE  MARTEDI  
ecc., ecc.
```

Si noti che dopo SPC(5) che spazia 5 battute dalla parola SETTEMBRE, si è dovuto fare il ; per far scrivere LUNEDI di seguito e non a capo come avverrebbe se il ; fosse omesso.

Notizie di informatica

LINGUAGGI

Siccome il calcolatore capisce soltanto la lingua dei bytes, per potere dialogare con questa prodigiosa macchina è indispensabile servirsi di programmi di traduzione chiamati anche linguaggi. Questi dovranno venire caricati nella memoria del computer prima di iniziare il lavoro. Taluni contengono già il linguaggio dentro di sé, caricato in maniera fissa in una memoria ROM (a sola lettura): è il caso, ad esempio, del Commodore 64 che ha il Basic Interpreter al suo interno tanto che appena acceso il computer, è possibile iniziare il lavoro senza altro indugio.

Elenco dei linguaggi più usati nella compilazione di programmi per computer:

COBOL

È adatto per programmi di gestione aziendali. È strutturato in modo da gestire una grande quantità di cifre e di calcoli.

PASCAL

È stato messo a punto per il lavoro scientifico.

FOR TRAN

È il primo linguaggio adoperato dai programmatori.

È la sigla di FORMula TRANslator (Traduttore di formule).

ALGOL

Altro linguaggio strutturato.

BASIC

Linguaggio interprete (traduce istruzione per istruzione) adatto per iniziare a programmare. Si dice che è ad "alto livello" perchè l'operatore dialoga direttamente con il computer. Dal punto di vista operativo presenta lacune sulla velocità di esecuzione dei programmi più complessi.

ASSEMBLER

È il linguaggio più vicino alla macchina in quanto servendosi delle sue sigle (MOVE-ADD-ecc.), l'operatore sposta i bytes passo per passo lungo il percorso operativo. Ha il notevole vantaggio di permettere una esecuzione velocissima in quanto col solo tramite di un assembler, queste istruzioni vengono trasformate in bytes che la macchina intende perfettamente.

COMPILATORE

È un programma-tramite col quale un qualsiasi programma di lavoro viene trasformato in bytes di istruzioni.

MEMORIA RAM

È la memoria di lavoro del computer. Tipica la capacità di memoria di 64 K bytes (64.000 bytes circa) del Commodore 64. Quando la macchina viene spenta, la memoria RAM perde tutti i suoi dati. È per questo motivo che per non cancellare i programmi e i dati forniti durante il lavoro esecutivo, questi vengono trasferiti in memorie magnetiche esterne (nastro di cassetta oppure floppy disk), mentre le istruzioni del programma operativo interno, dell'EDITOR (gestione del video) e del BASIC INTERPRETER (in quei computer che l'hanno in permanenza), provengono da memorie ROM interne al calcolatore.

Queste istruzioni non possono essere distrutte nè cambiate.

INDIRIZZI

Ogni qualvolta un byte (dato o istruzione) viene introdotto (scrittura) oppure estratto (lettura) dalla memoria RAM, occorre preventivamente attivare la striscia di memoria corrispondente a quel determinato byte. (Fig. 10). Per

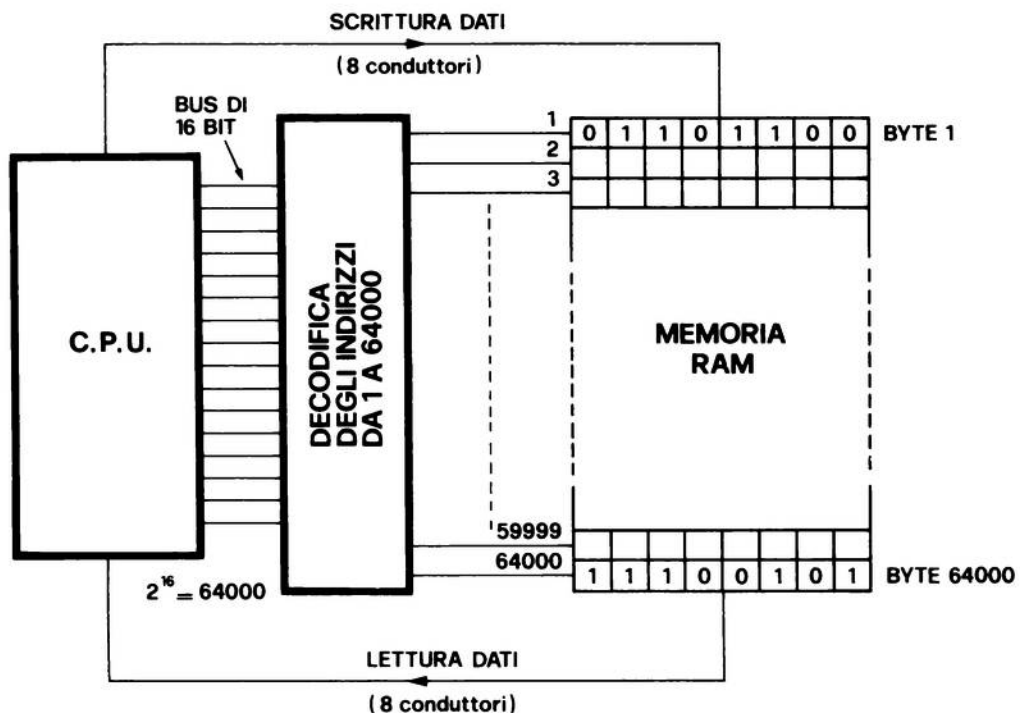


Fig. 10

raggiungere le 64.000 strisce di una memoria RAM occorrono 64.000 numeri di indirizzo. Per arrivare a queste 64.000 locazioni, occorre un decodificatore al cui ingresso deve essere fornito un numero da 1 a 64.000.

Tutte queste combinazioni si possono ottenere con una parola binaria di 16 bits.

Infatti, con 16 bits si hanno 2^{16} combinazioni = 65.536, arrotondati in 64 kbytes per comodità d'espressione.

CAPACITA' DI MEMORIA

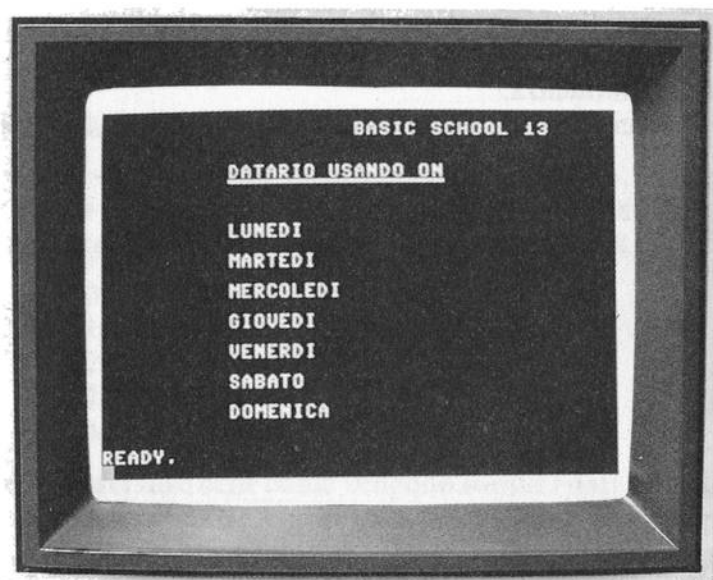
Vi sono memorie RAM di 8-16-32-64-128-256 ecc. kilo bytes. Una memoria più ampia accresce la velocità di esecuzione in quanto può caricare programmi complessi ed avere ancora liberi una grande quantità di K bytes per il lavoro.

Tuttavia, non sempre il costo, che cresce notevolmente con l'incremento della memoria RAM suggerisce l'acquisto di computer con RAM elevata. La maggior parte dei programmi è suddivisa in tanti sottoprogrammi i quali vengono caricati separatamente e nella sola quantità necessaria a svolgere il lavoro richiesto in quel momento. Ciò naturalmente rallenta l'esecuzione di qualche secondo, ma la diminuita spesa d'acquisto spesso compensa questa piccola perdita.

Lezione pratica

ON...GOTO

(all'espressione ON...GOTO si riferisce l'esercizio n. 13)



**Esercizio n. 13
RUN completo.**

Letteralmente significa SU...VAI A ed ha questo significato specifico: su ciascun numero attribuito alla variabile numerica preso con andamento pro-

gressivo, salta alla riga di programma indicata con la stessa progressione.
Esempio:

```
100 INPUT "1. INTRODUZIONE DATI" ; P
110 INPUT "2. CONSULTAZIONE DATI" ; P
120 INPUT "3. AGGIORNAMENTO DATI" ; P
130 ON P GOTO 300,500,700
300 PRINT "INTRODUZIONE DATI": END
.....
.....
.....
.....
.....
500 PRINT "CONSULTAZIONE DATI" : END
.....
.....
.....
.....
700 PRINT "AGGIORNAMENTO DATI" : END
```

Se al primo INPUT rispondo con il numero 1 il programma salta alla riga 300, se rispondo 2 salta alla riga 500, ecc.

L'espressione ON...GOTO ha una funzione semplificativa rispetto alla funzione IF...THEN e va utilizzata in particolare negli INDICI, detti anche MENU', che stanno all'inizio di un complesso di più sottoprogrammi.

Se si risponde con un numero diverso da quelli posti nell'indice, l'istruzione ON...GOTO non ha alcuna influenza.

Per la spiegazione dell'esercizio n. 13 andare alla lettura del listing corrispondente.

BASIC SCHOOL - Lezione n. 12

Lezione pratica

READ...DATA (aperto)

Questa espressione viene utilizzata quando si debbano far leggere al calcolatore una grande quantità di DATI (NUMERICI O STRINGA) con una sequenza prestabilita. Es.:

```
100 READ P (variabile numerica)
110 DATA 5,43,36,28.5,1432,75,1
120 PRINT P
130 GOTO 100
```

N.B. Ricordare: a) che dopo l'istruzione DATA non va posta la virgola b) che fra un dato e l'altro sia numerico che letterale va interposta una virgola di separazione c) che nei dati numerici non interi, la virgola va fatta con il punto (.) all'inglese.

In questo caso, con la riga 100 comandiamo al computer di leggere la variabile numerica cui attribuiamo la sigla P. Viene letto come primo dato il n. 5, quindi viene stampato il suo valore quando il programma arriva alla riga 120. Dopo di che il programma ritorna alla riga 100 dove riceve l'istruzione di leggere il DATO seguente, cioè il valore 43. Questo ciclo di lettura va avanti fintanto che esistono dati da leggere. Al termine, il calcolatore presenta sullo schermo la frase: **OUT OF DATA,ERROR** la quale sta a significare che non vi sono più dati a disposizione.

READ...DATA (chiuso)

Per ovviare a questo inconveniente si possono limitare i cicli di lettura al numero strettamente necessario per leggere i dati a disposizione. In questo caso, il computer non segnala errori di sorta. Es.:

```
100 FOR K = 1 TO 6
110 READ P
120 DATA 35, 24, 32.8, 76, 5, 1
130 PRINT P
140 NEXT K
```

N.B. Osservare bene come vengono forniti i dati da leggere: dopo l'istruzione DATA essi vanno elencati in sequenza progressiva separati fra loro da una virgola. È da notare che la virgola che separa i decimali dagli interi, viene segnata con un punto, come sopra indicato per i numeri 28.5 e 32.8

READ...DATA (stringa).

L'espressione READ...DATA può essere utilizzata anche per leggere dei dati

letterali o misti (dati STRINGA). È il caso ad esempio di un programma che intenda stampare un indirizzo come questo:

```
100 FOR K =1 TO 5
110 READ I$
120 DATA COGNOME, NOME, VIA E NUMERO, CAP. -
      LOCALITA', PROVINCIA
130 PRINT TAB(8) I$
140 PRINT
150 NEXT K
```

L'esecuzione vedrà scritte una sotto l'altra, spaziate di una battuta (riga 140) le frasi:

```
COGNOME
NOME
VIA E NUMERO
CAP. - LOCALITA'
PROVINCIA
```

Seguendo l'esercizio 14

```
110 PRINT TAB(9) "SETTEMBRE" G$
```

Ad ogni passaggio, viene scritta la frase SETTEMBRE e, accanto, il nome della variabile stringa G\$ letta in quel passaggio (LUNEDI, MARTEDI, ecc.).



Esercizio n. 14
RUN completo.

Seguendo l'esercizio 15

70 FOR K = 18 TO 24

Qualora si intenda sfruttare la variabile di ripetizione (ciclica) non soltanto per indicare il numero dei cicli, ma anche per indicare un numero proprio (qui sta a significare i giorni di quella settimana), K può anche non partire da 1 come spesso si fa in questi casi.

```
110 PRINT TAB(6) K "SETTEMBRE -" SPC(2) G$
```

Si tratta di un PRINT triplo, uno accanto all'altro. Dapprima, tabulando 6 battute, viene scritto il valore di K (da 18 a 24, i giorni del mese di quella particolare settimana), quindi la frase "SETTEMBRE -" poi, spaziando 2 battute sulla destra, il nome della variabile stringa G\$ letta ad ogni passaggio. Es.:

```
18 SETTEMBRE - LUNEDI  
19 SETTEMBRE - MARTEDI  
20 SETTEMBRE - MERCOLEDI  
ecc. ecc.
```

SPC(2)

La sigla SPC significa SPACE cioè SPAZIO e serve per spaziare fra di loro due PRINT in senso orizzontale. Con questo, sono 3 i sistemi possibili per spostare a destra il cursore.



Esercizio n. 15
RUN completo.

PRINT,, “FRASE”

Le virgole tabulano automaticamente a destra il cursore, ciascuna di 10 battute (max concesso, 3 virgole).

PRINT TAB(6) “FRASE”

Viene tabulato l'inizio della FRASE partendo dal margine sinistro della pagina, di tante battute quanto è il numero contenuto fra le parentesi (in questo caso si tabula di 6 battute). Ricordare che TAB(23) va scritto **tutto di seguito senza spazi intermedi**.

PRINT TAB(10) “FRASE 1” SPC(3) “FRASE 2”

In questo caso, dopo la prima frase scritta tabulando 6 battute, viene lasciato uno spazio di 3 battute e poi scritta la seconda frase.

BASIC SCHOOL - Lezione n. 13

Lezione pratica

Eeguire il RUN dell'esercizio n. 16.



Esercizio n. 16
RUN completo.

Si tratta di un READ...DATA con numero chiuso da un FOR...NEXT che comprende 10 cicli (Fig. 11). Ad ogni passaggio viene scritto il valore della variabile numerica, letta nel DATA, immediatamente successiva (riga 80), quindi viene fatta la sommatoria dei dati fin qui letti con $S=S+P$.

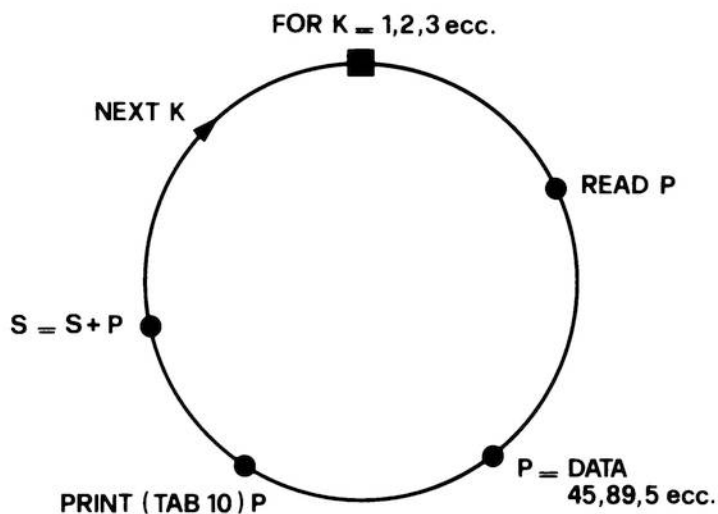


Fig. 11

Fuori dell'operazione ciclica di lettura viene scritta la somma di tutti i dati letti e il loro valore medio, eseguendo l'operazione $S/(K-1)$. (N.B. Il tasto di divisione è / e non i due punti).

Ricordare che quando K va da 1 a 10, esso assume 10 valori progressivi, ma quando esce dall'ultimo ciclo o loop il suo valore è 11, quindi per fare la media occorre utilizzare il valore $K-1$ che è il numero di DATI passati in lettura. Ciò, naturalmente vale soltanto quando K inizia da 1 e ha un andamento continuo, cioè senza STEP.

Notizie d'informatica

IL PERSONAL COMPUTER

Il personal computer è il fratello maggiore dell'home computer (calcolatore per famiglia). Esso presenta rispetto a quest'ultimo una maggiore completezza. Infatti si compone di una tastiera (che talvolta è separata), di un elaboratore centrale, di un monitor altamente professionale, di 2 lettori di floppy disk e di una stampante. Per quanto riguarda i programmi applicativi questi dipendono dal lavoro che si intende svolgere con questo tipo di computer che chiameremo "il calcolatore del professionista". Il personal è utilissimo per studi professionali, artigiani, piccole aziende, in quanto permette con l'impiego di 2 o 3 programmi la completa gestione della propria attività.

I programmi tipici per questa macchina sono:

1. Il Word Processing

Per la gestione completa di testi, lettere, indirizzi. È possibile stampare lettere pescando indirizzi diversi e con le qualità grafiche preferite. È possibile stampare ad alta velocità con una stampante ad aghi bidirezionale oppure a bassa velocità con una macchina da scrivere elettronica a margherita purchè sia interfacciata con il computer.

2. Amministrazione generale

È sufficiente un unico programma amministrativo per tenere tutta la contabilità di una piccola azienda (calcolo dell'IVA, elenco fornitori, elenco clienti, gestione del conto corrente bancario, rendimenti finali, ecc.).

3. Magazzino

Per quelle aziende che hanno un certo movimento di magazzino, vi sono programmi generali o personalizzati che permettono ad un solo operatore di aggiornare in tempo reale la situazione dei pezzi presenti in stoccaggio.

Il Triumph Adler P2

In questo breve corso di programmazione, il personal computer viene presentato al solo scopo conoscitivo per ampliare e completare le notizie di informatica del lettore.

Il personal TRIUMPH-ADLER modello Alpatronic P2 è un computer

compatto che ha una memoria RAM di 48 Kbytes, 2 lettori di floppy disk da 5" con memoria di massa di 240 Kbytes per disco, un monitor a fosfori verdi con ottima risoluzione e focalizzazione. Può essere abbinato a qualsiasi tipo di stampante.

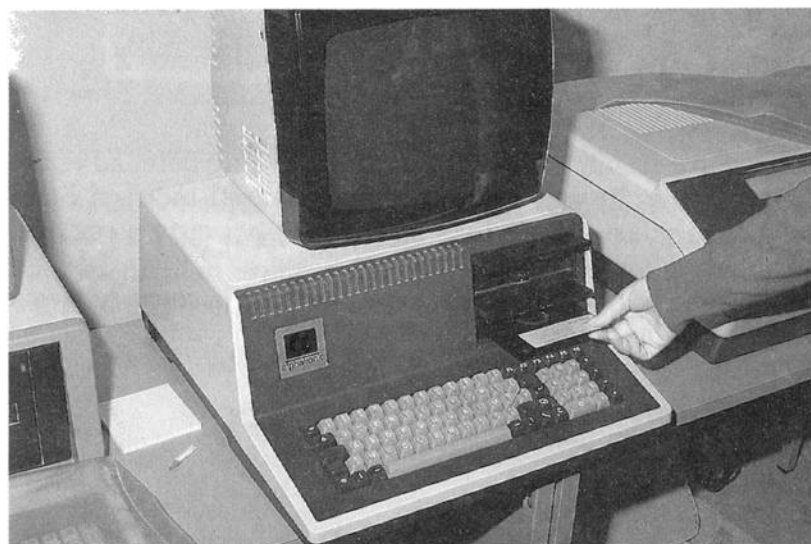
Utilizzo del Word Processing

L'utilizzo del P2 come gestore di testi e indirizzi è uno dei tipici impieghi di questa macchina assai versatile. Infatti, la sua tastiera contiene 6 tasti di funzioni (F1-F2-F3-F4-F5-F6) ed altri tasti aggiunti che sono stati appositamente progettati per la gestione dei testi.

Caricamento del programma

Inserire nella fessura inferiore il dischetto che contiene il programma word processing con l'etichetta rivolta verso l'alto e inserendo per prima la finestrella di lettura. Accompagnare delicatamente il disco fino alla fase di arresto quindi chiudere lo sportellino di chiusura. Inserire con la stessa procedura un disco nuovo nella fessura del lettore superiore, dopo aver chiuso con uno scotch apposito la tacca di protezione posta su un lato del disco.

Anche qui, al termine dell'operazione, chiudere lo sportellino.



Personal Computer TRIUMPH-ADLER P2. Ecco come va inserito il dischetto contenente il programma nel lettore di floppy inferiore. L'etichetta del disco deve trovarsi in alto, e dalla parte dell'operatore. Inserire senza forzare accompagnando il disco fino in fondo, poi chiudere lo sportellino.

Giunti a questo punto, occorre premere la lettera B e il tasto che corrisponde al tasto RETURN del Commodore.

Automaticamente si accenderà la spia del lettore (mangiadischi) che contiene il programma (segnale di lettura o scrittura in atto) e verrà caricato l'indice

di apertura. Come si può notare durante il corso delle operazioni, il programma non viene caricato al completo nella memoria RAM, ma è stato suddiviso in molti sottoprogrammi e di volta in volta viene caricata la parte che interessa per le operazioni del momento. In questo modo non si intasa la memoria di lavoro del calcolatore. È inoltre possibile, ai programmatori, apportare delle modifiche al programma stesso senza doverlo cambiare completamente.

Inizializzazione del dischetto

Comparirà un primo indice su cui scegliere il tipo di operazione da svolgere. Successivamente compariranno altri indici su cui effettuare altre scelte più selettive. In questo caso va scelta l'INIZIALIZZAZIONE (o FORMATTAZIONE) operazione che verrà svolta sull'unità 2 cioè il disco posto nel lettore superiore. Al termine di questa operazione che consiste nel predisporre le piste e i settori magnetici sul disco, tutto è pronto per l'impiego corrente del computer.

Word processing

Come già detto, col programma Alphatest (word processing) si possono scrivere e memorizzare lettere, articoli, ed ogni altro tipo di testo. Vi si possono effettuare ogni tipo di correzioni, aggiunte, cancellazioni, ecc. fino ad ottenere un testo perfetto che verrà memorizzato sul dischetto 2 fornendogli preventivamente una sigla di identificazione che dovrà essere ripetuta tale e quale (caratteri e spazi) durante il richiamo da memoria.

È poi possibile archiviare grandi quantità di indirizzi (o altre cose, ad esempio libri) contenenti fino a 20 voci significative. Da questi indirizzi è poi possibile fare delle estrapolazioni selettive, ad esempio per città o regione, ecc.

Come accennato, durante la fase di stampa si può scrivere la stessa lettera a più indirizzi diversi.

BASIC SCHOOL - Lezione n. 14

Lezione pratica

OPERAZIONE CICLICA DOPPIA (DOPPIO LOOP)



Esercizio n. 17
RUN completo.

Talvolta, nel corso di un'operazione FOR...NEXT si inserisce un secondo LOOP (anello) come è indicato nell'esercizio n. 17 a cui facciamo riferimento. Nel disegno esplicativo, ogni valore di K, quando giunge all'anello J, viene moltiplicato per tutti i valori di J (fig. 12).

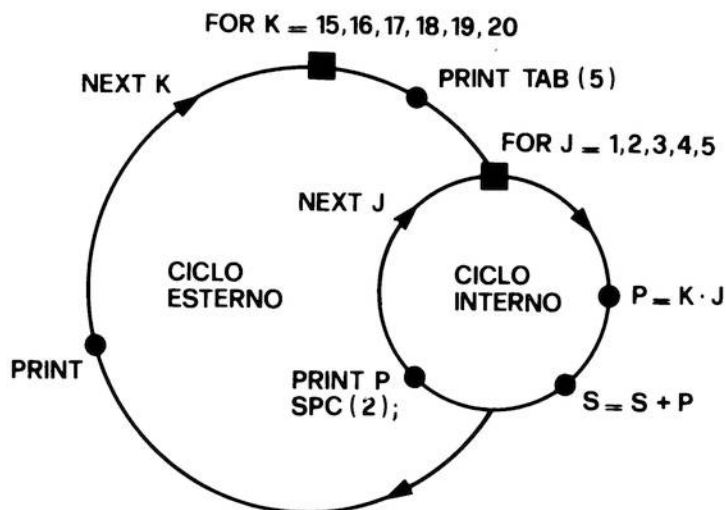


Fig. 12

Così, ad esempio, quando $K = 15$, giunto al ciclo interno viene moltiplicato per 1,2,3,4,5. Esce poi dal ciclo interno e riprende il suo percorso nel ciclo esterno. Giunto al NEXT K, K assume il nuovo valore di 16 e così via. Una volta effettuato il disegno del DOPPIO LOOP occorre stabilire le operazioni matematiche, le frasi da scrivere, gli spazi da lasciare sia per quanto riguarda le operazioni del ciclo J sia per ciò che concerne le operazioni del ciclo principale, quello di K.

Nell'esercizio 17, queste operazioni sono:

```
95 PRINT TAB(5).
```

```
Tabula di 5 battute prima di eseguire il ciclo interno
```

```
110 P = K * J
```

Questa operazione va inserita all'interno del ciclo J in modo che ogni valore di K sia moltiplicato per tutti i valori di J

```
120 S = S+P
```

Per effettuare la sommatoria dei prodotti occorre che il calcolo $S=S+P$ venga effettuato ad ogni ciclo interno, fino ad assumere il valore finale.

```
130 PRINT P SPC(2) ;
```

Anche questa operazione come le precedenti va inserita nel ciclo interno. Significa: scrivi il valore di P, poi spazia 2 battute e ferma il cursore.

Chiusura dei cicli

```
140 NEXT J
```

Chiusura del ciclo interno. Va fatta per prima.

```
145 PRINT
```

Serve per rimandare a capo il cursore quando esce dal ciclo interno.

```
150 NEXT K
```

Chiusura del ciclo esterno. Va fatta per ultima.

Fuori dal doppio loop

```
170 PRINT TAB(2) "LA SOMMA DI TUTTI I PRODOTTI È" S
```

```
190 PRINT TAB(2) "LA MEDIA DI TUTTI I PRODOTTI
```

```
È" S/30
```

Per calcolare il valore medio occorre sapere quanti sono i prodotti P nati da questa doppia operazione ciclica.

Quando invece sia K che J siano progressivi, la soluzione di questo problema è abbastanza facile.

Chiamando K_f il valore finale di K

K_i il valore iniziale di K

J_f il valore finale di J

J_i il valore iniziale di J

$$N \text{ (numero dei prodotti)} = (K_f - K_i + 1)(J_f - J_i + 1)$$

Nel nostro caso: $N = (20 - 15 + 1)(5 - 1 + 1) = 30$

Da cui si ricava il valore medio $M = S/N = S/30$.

Qualora uno o entrambi i cicli siano rovesciati (es.: FOR K = 20 TO 15) STEP -1 oppure FOR J = 5 TO 1 STEP -1, la formula va modificata in questo modo:

$$N \text{ (numero prodotti)} = (K_i - K_f + 1)(J_i - J_f + 1)$$

Le difficoltà aumentano quando il conteggio non sia progressivo ma proceda a salti (STEP 2, STEP 5, STEP-3, STEP-2 ecc.).

Formula per progressione positiva

$$N = \left[\text{INT} \frac{K_f - K_i}{n. \text{ step } K} + 1 \right] \cdot \left[\text{INT} \frac{J_f - J_i}{n. \text{ step } J} + 1 \right]$$

Es.: per K = 1 TO 20 STEP 3
per J = 5 TO 15 STEP 2

$$N = \left[\text{INT} \left(\frac{20 - 1}{3} \right) + 1 \right] \cdot \left[\text{INT} \left(\frac{15 - 5}{2} \right) + 1 \right] = 7 \cdot 6 = 42$$

Formula per progressione negativa

$$N = \left[\text{INT} \left(\frac{K_i - K_f}{n. \text{ STEP } K} \right) + 1 \right] \cdot \left[\text{INT} \left(\frac{J_i - J_f}{n. \text{ step } J} \right) + 1 \right]$$

N.B. Nella formula per la progressione negativa il numero di STEP va posto con segno positivo. La sigla INT significa che del risultato di questa divisione va preso soltanto il numero INTERO, trascurando quindi eventuali decimali.

BASIC SCHOOL - Lezione n. 15

Lezione pratica

(con riferimento agli esercizi n. 18 e 19)

Le matrici lineari

I dati, sia numerici che del tipo stringa, trattati con i sistemi finora conosciuti (FOR...NEXT — READ...DATA) non si prestano molto alla loro elaborazione in quanto una volta letti vengono dimenticati dalla memoria del calcolatore. Si salvano soltanto i risultati finali delle operazioni matematiche indicate nel corso del programma. Esiste inoltre una difficoltà, quando si abbiano in INPUT una grande quantità di dati, per attribuire a ciascuno di essi una variabile che serva alla singola identificazione. Ciò è particolarmente vero quando questi dati debbano venir salvati su disco o cassetta e di seguito ripescati per la loro consultazione o per il loro aggiornamento.

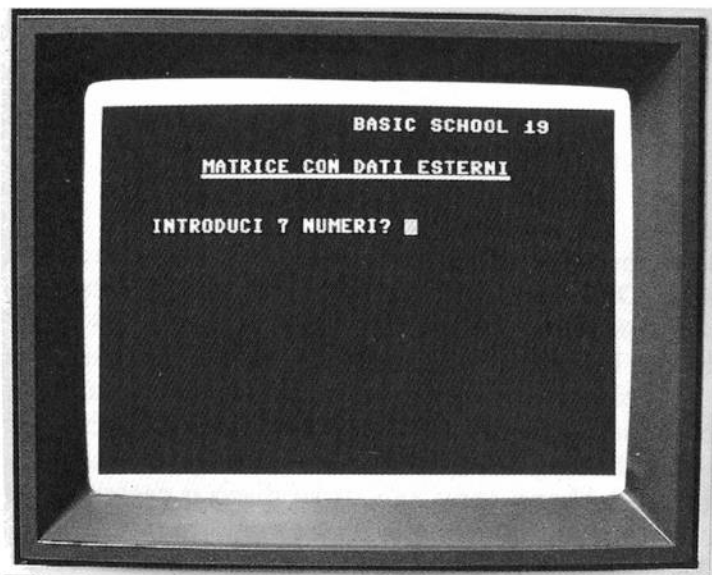


Esercizio n. 18
RUN completo.

Con l'introduzione delle matrici, tutti i dati tra di loro omogenei vengono trattati con una sola variabile con indice numerico (es.: $D(I)$ dove D sta per dato e I per indice).

Con questo sistema, i dati forniti con l'INPUT al calcolatore, vengono memorizzati in caselle di memoria numerate progressivamente dal n. 1 al numero finale dei dati stessi. Esempio:

```
10 FOR I = 1 TO 5
20 READ D (I)
30 DATA 24, 32, 16, 3, 4
40 PRINT D(I);
50 NEXT I
```

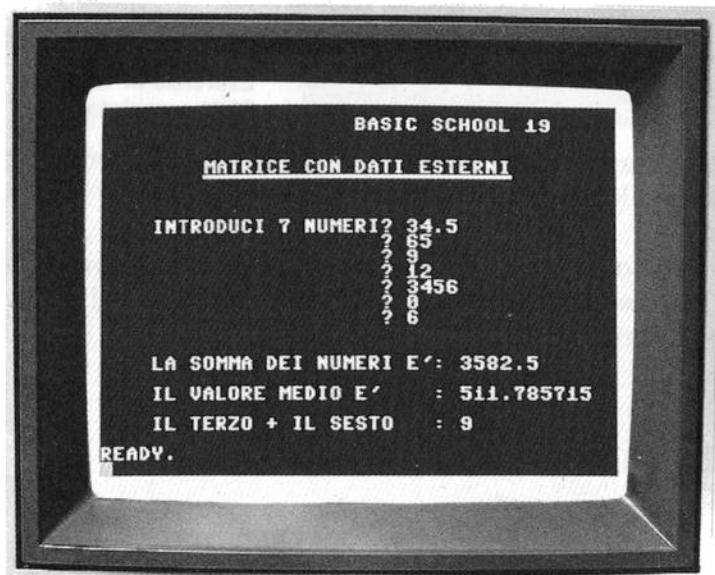


Esercizio n. 19
RUN fermo
al primo INPUT.

Con questo programma elementare, il calcolatore legge i 5 dati forniti, li scrive uno accanto all'altro e li tiene in memoria attribuendo l'indice 1 al primo dato (24), l'indice 2 al secondo dato (32), ecc. Riassumendo:

- D(1) = 24
- D(2) = 32
- D(3) = 16
- D(4) = 3
- D(5) = 4

Questo sistema di classificazione dei dati forniti (dall'interno del programma oppure dall'esterno con un INPUT progressivo) permette ora di elaborare i dati secondo le necessità. Esempio: Se desidero sommare il 1° con l'ultimo



Esercizio n. 19
RUN completo.


```
90 PRINT TAB(17)
100 INPUT D$(I)
110 NEXT I
120 OPEN 1,8,1,"INDIRIZZO"
130 FOR I = 1 TO 5
140 PRINT# 1, D$(I)
150 NEXT I
160 CLOSE 1
```

Avrete capito che si tratta di un indirizzo che alla fine viene salvato su disco. La prima parte del programma, fino alla riga 60 stampa su schermo l'enunciato o maschera, cioè COGNOME, NOME, ecc.

Dopo di che il cursore viene fatto ritornare all'altezza della parola COGNOME, e vengono effettuati cinque INPUT, ogni volta TAB(17) per far uscire i punti di domanda a destra del PRINT.

In questo modo vengono forniti 5 dati stringa D\$(I) indicizzati (es.: ROSSI PIERO, ecc.) che dalla riga 120 in poi vengono salvati su disco fino alla chiusura del file.

N.B. Non ci si deve dimenticare che nell'espressione PRINT # 1 va lasciato uno spazio fra # e 1.

Conclusione

Le operazioni che si effettuano sulla variabili con indice sono le stesse che si possono effettuare con le variabili normali; sola differenza sostanziale è che nelle matrici la variabile va sempre indicizzata: D(I) - D\$(I) ecc. e che durante la lettura dei dati con il FOR...NEXT, la variabile ciclica è di solito indicata con la I (perchè ricorda la parola Index=Indice): FOR I = 1 TO N, ecc. dove N è il numero complessivo dei dati introdotti.

Lezione pratica

Creazione della maschera

Si osservi il RUN dell'esercizio n. 20. Si noterà che a differenza di quanto è avvenuto finora, l'elaborato o PRINTATO del programma compare davanti all'operatore in tutta la sua completezza cosicchè egli ha subito un'idea esatta dei compiti che gli stanno davanti. Nello stesso tempo, vedrà il punto interrogativo e il cursore pulsante del primo INPUT restare in attesa del dato che dovrà essere fornito dall'esterno. Una volta soddisfatta questa esigenza, il cursore si sposterà sul secondo INPUT e così via fino al termine del programma. Qui giunti, c'è in genere una scelta operativa tra il chiudere il programma oppure ripeterlo oppure ritornare ad un indice iniziale quando quel determinato programma sia strutturato in sottotitoli.



Esercizio n. 20
RUN fermo
al primo INPUT.
Come si può notare,
il printato è già
al completo prima ancora
che arrivi
il 1° INPUT.

Ritorno in alto del cursore

Per avere la possibilità di svolgere graficamente il programma degli INPUT sull'intero printato o "maschera" preparato in precedenza, dapprima si scrivono sullo schermo tutte le frasi fisse (titoli - sottotitoli - frasi esplicative -ecc.), quindi si riporta artificialmente il cursore in alto, nella posizione del 1° INPUT.

Vengono poi posizionati i successivi INPUT fino all'ultimo.

Posizionamento del cursore con PRINT CHR\$; e PRINT TAB(X)

Per riposizionare il cursore una volta stampato l'elaborato, occorrono due istruzioni: una di ritorno verso l'alto e una seconda di spostamento a destra per occupare la posizione utile per il 1° INPUT.

Es.: 130 FOR K= 1 TO 17: PRINT CHR\$(145) ; : NEXTK

con questa espressione, si riporta in alto il cursore di 17 battute, una di seguito all'altra in funzione dell'azione del punto e virgola (;) posto dopo il PRINT CHR\$(145). Ricordiamo che (145) è il codice interno al Commodore 64 che indica CURSORE VERSO L'ALTO, mentre PRINT CHR\$ (si legge PRINT CHARACTER STRINGA) è l'istruzione esecutiva di questa specifica funzione.

140 PRINT TAB(27)

Dopo aver portato con la riga 130 il cursore verso l'alto, con questa seconda espressione, lo spostiamo verso destra, appena al di fuori della prima frase esplicativa che naturalmente riguarda un dato da fornire dall'esterno oppure da trarre internamente dallo stesso calcolatore (esempio: prodotto, somma, media aritmetica, ecc.). Dopo di che procediamo a creare il 1° INPUT con

150 INPUT N.

Finita questa prima operazione di INPUT, riposizioneremo di nuovo il cursore con eventuali PRINT di spaziatura verticale e PRINT TAB(X) di tabulazione orizzontale (X è il numero delle battute) e creeremo i successivi INPUT.

Posizionamento del cursore col sistema grafico

Esiste un secondo sistema per posizionare il cursore dopo il completamento della maschera, quello di fornire l'istruzione INPUT, posizionando il cursore con segni grafici interni alla virgolette.

Es.: INPUT "◐◑◒◓◔◕◖◗◘◙◚◛◜◝◞◟◠◡◢◣◤◥◦◧◨◩◪◫◬◭◮◯◰◱◲◳◴◵◶◷◸◹◺◻◼◽◾◿◿"; N

I segni grafici interni alle virgolette sono stati effettuati, il primo premendo i tasti SHIFT + CURSR ↑ (cursore verso l'alto) e il secondo premendo il tasto CURSR → (cursore a destra).

Il punto interrogativo di questo INPUT comparirà quindi, rispetto all'ultima posizione di riposo, spostato di 8 battute più in alto e 9 battute verso destra.

Commento all'esercizio n. 20

70.

Il numero dei dati da fornire dall'esterno non deve superare la decina. Qui, per ragioni di spazio. Tuttavia quando si superano i 10 dati, occorre preventivamente dimensionare le matrici. Es.:

```
3 DIM D (15)
220 FOR I= 1 TO 15
230 READ D(I)
240 DATA 6,56,3,9.8,0,54,768,23,12,89,987,2.9,12,1,9
250 PRINT TAB(10) D(I)
260 NEXT I
```

N.B. Il dimensionamento della matrice va effettuato all'inizio del programma prima di qualsiasi altra operazione.

10-120

Sono state scritte tutte le frasi fisse che compongono la cosiddetta MASCHERA. Eventualmente, provare il RUN di questo esercizio, una volta arrivati alla riga 120.

130-140

Riposizionamento del cursore per introdurre il 1° INPUT.

150-200

Introduzione degli input che riguardano il numero dei dati N e gli stessi dati D(I). Questi ultimi sono stati preventivamente tabulati di 16 battute.

220-240

Per posizionare l'INPUT della riga 270 sempre nello stesso punto, considerato che il numero delle righe occupate dai dati dipende dal valore che si dà a N (INPUT in successione verticale), non c'era altro mezzo che riportare il cursore in posizione HOME (in alto a sinistra) e poi scendere di 18 battute, quindi tabulare di 28 battute.

N.B. Per portare il cursore in posizione HOME, scrivere PRINT "S", mettendo fra le virgolette il segno grafico che si forma premendo il tasto HOME da solo.

280-290

Condizione per chiudere il programma oppure ripeterlo. Qualsiasi risposta che non sia la S, chiude il programma stesso.

Nota bene

Interessanti sono le righe n. 160 e n. 210. Con la riga 160, qualora alla domanda QUANTI DATI? venga fornito come risposta lo 0 (zero) oppure venga battuto il RETURN, la somma PU deve risultare uguale a zero e non deve accettare dati dall'esterno (salta direttamente alla riga 220). Con la riga 240, qualora N sia uguale a 1, $PU=D(1)$. Senza questa riga di programma, PU sarebbe uguale al doppio di D(1) perchè nella circostanza, D(1) sarebbe anche D(N) e il dato verrebbe conteggiato due volte (riga n. 215).

Lezione pratica

L'esercizio n. 21 (Fig. 13) è un vero programma completo di gestione dati, seppure in miniatura.

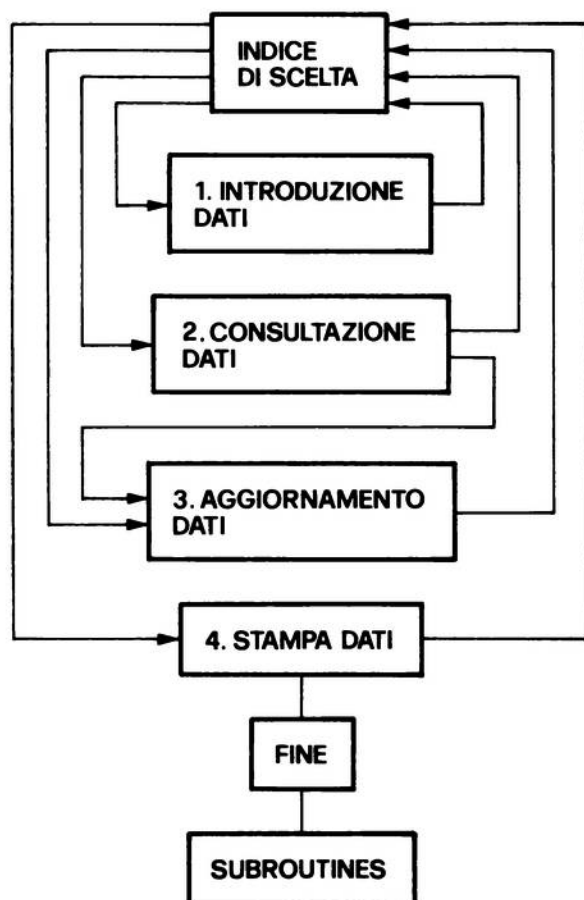


Fig. 13

Questo esercizio consta di un indice o menù, dal quale si accede a 4 sottoprogrammi di gestione dei dati numerici. Col sottoprogramma n. 1 vengono introdotti ed eventualmente salvati su cassetta i dati, col n. 2 essi possono venir recuperati dalla cassetta e consultati, col n. 3 i dati consultati possono essere aggiornati e salvati di nuovo. Con il n. 4, i dati possono venire stampati su carta per mezzo di una normale stampante per computer. In coda al programma vi è poi una serie di subrotine che vengono richiamate secondo necessità.

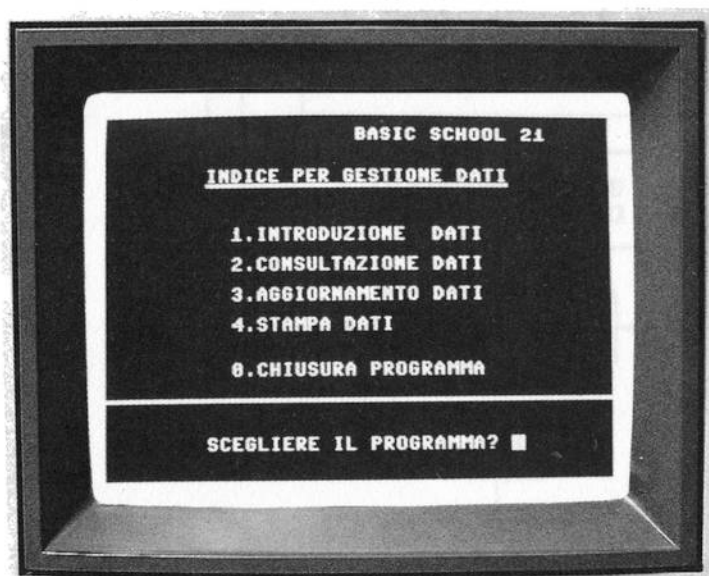
Esercizio 21 - Prima parte

Creazione dell'indice e delle subroutines

L'esercizio n. 21 (vedere il listato completo al termine della Lezione n. 19) riguarda la gestione dei dati su cassetta. Si compone di un indice per scegliere il tipo di elaborazione che si intende svolgere sui dati stessi, e di 4 sottoprogrammi intitolati:

1. INTRODUZIONE DATI (con salvataggio su cassetta)
2. CONSULTAZIONE DATI (dopo averli prelevati dalla cassetta)
3. AGGIORNAMENTO DATI (e salvataggio di quelli che vengono cambiati e memorizzati al posto di quelli non più utili).
4. STAMPA DATI.

Al termine del quadruplo programma, vengono predisposte le subroutines dei microprogrammi che vengono ripetuti qua e là durante tutto l'esercizio. Es.: sub. per spaziare in verticale, per ritornare in alto col cursore, per tabulare, per la maschera del programma, ecc.



Esercizio n. 21
Presentazione dell'indice
con scelta
dei sottoprogrammi.

Naturalmente, dall'indice si accede a tutti e quattro i sottoprogrammi. Da ciascuno di questi, premendo la lettera N (NO) si ritorna all'indice, dal quale si può poi accedere ad un altro sottoprogramma oppure chiudere premendo lo 0 (zero). Ogni sottoprogramma chiude con una scelta fra S e N. Se la scelta è positiva (S) verrà svolta l'operazione dichiarata dalla frase che precede S/N.

Scelta S/N

Così ad esempio, nel sottoprogramma 1 che serve per introdurre i dati dall'esterno, se al termine si preme la lettera S, viene attivata quella parte di listato che salva su cassetta i dati forniti e/o calcolati.

Naturalmente l'uso della cassetta al posto del lettore di floppy disk, comporta qualche lentezza e qualche difficoltà iniziale. Tuttavia, lo scopo didattico, viene ugualmente raggiunto, ed è quello di effettuare un programma realistico al 100%.

1ª parte - Indice e subroutines

La compilazione dell'indice andrebbe fatta al termine di tutto l'esercizio 21, perchè non si sa a priori a quale linee di programma mandare il calcolatore quando si sceglie il sottoprogramma (1 o 2 o 3 o 4). Tuttavia, per facilitare la comprensione di questo esercizio, esso viene scritto appena dopo aver programmato le subroutines, le quali essendo dei miniprogrammi che vanno ripetuti un po' dappertutto, devono precedere ogni altro lavoro, se si vogliono alleggerire le altre operazioni.

Subroutines

Le subroutines devono venir poste al termine di tutto il programma dopo che questo è stato chiuso con END.

Non sapendo a priori la lunghezza finale del listato, abbiamo deciso di partire da 5000 chiudendo con END su questa riga tutto ciò che sta in precedenza. Occorre scriversi a parte i numeri di riga che riguardano le subroutines. Ricordarsi di porre dei REM esplicativi all'inizio di ogni subroutine e di chiudere ciascuna di esse con l'espressione RETURN.

Le subroutines preparate riguardano:

Spaziare in verticale

```
5010 FOR K = 1 TO SP : PRINT : NEXT K : RETURN
```

Le subroutines 5010 serve per spaziare in verticale. Basterà nel corso del programma dare un valore ad **SP** (spaziare) e scrivere GOSUB 5010. Es.:

```
30 SP = 3 : GOSUB 5010 (per spaziare tre battute in verticale)
```

Separare con linea orizzontale

```
5030 FOR K = 1 TO 40 : PRINT "☐"; : NEXT K : RETURN
```

La subroutine 5030 serve per stampare una linea di separazione orizzontale larga quanto lo schermo. Es.:

```
140 GOSUB 5030
```

Ritornare verso l'alto

```
5050 FOR K = 1 TO R : PRINT CHR$(145); : NEXT K : RETURN
```

La subroutine 5050 serve per ritornare in alto col cursore. Basterà scegliere prima la **R** (Ritornare) che sarà un numero equivalente alle battute di ritorno richieste. Es.:

```
330 R = 17 : GOSUB 5050 (per ritornare in alto, col cursore, di 17 battute)
```

Tabulare

5070 PRINT TAB(T) : RETURN

La subroutine 5070 serve per tabulare il cursore verso destra. Basterà fornire prima il valore di T (tabulare) che corrisponde al numero delle battute di tabulazione richiesto. Es.:

410 T = 28 : GOSUB 5070 : PRINT N

Significa che il cursore va a destra di 28 battute e poi stampa il valore di N.

Maschera ripetuta

5090 - 5160

È la subroutine che serve per printare le frasi fisse da "QUANTI DATI? MAX (10):" fino alla linea orizzontale di separazione prima della scelta finale. Siccome questa parte si ripete nei programmi 1, 2 e 3, è stata per essa predisposto questo miniprogramma.

Salvataggio dati su cassetta

5235 - 5290

Con questo miniprogramma, viene aperto il file n. 1 per travasare i dati su cassetta. Questi dati sono quelli rappresentati dalle variabili N, D(I) e PU. L'espressione che indica la scrittura sulla periferica di memoria è:

PRINT# 1, Variabile

dove 1 è il numero del file precedentemente aperto. Ricordare che tra # e 1 va lasciato uno spazio. Al termine del travaso di dati, il file viene chiuso con l'espressione CLOSE 1.

Da notare, infine, che la prima istruzione di questa sub. è: 5235 PRINT. Quando il calcolatore legge la OPEN, stampa automaticamente sullo schermo la frase PRESS RECORD AND PLAY ON TAPE (premi REGISTRAZIONE ed AVVIO sul registratore). Senza il PRINT della riga 5235, questa frase verrebbe scritta sopra quella che precede la scelta S/N.

Recupero dati da cassetta

5340 - 5410

Questa subroutine rappresenta l'operazione contraria di quella descritta qui sopra. Qui si tratta, infatti, di leggere i dati dalla cassetta per caricarli nella memoria RAM del calcolatore per poterli consultare ed eventualmente aggiornare.

Le operazioni sono sostanzialmente simili a quelle descritte per il salvataggio dati, salvo per il fatto che in questa occasione, al posto dell'espressione PRINT# viene utilizzata quella di lettura INPUT# e che nell'apertura del file, al posto di OPEN 1,1,1 va scritto OPEN 1,1,0 (lo 0 significa lettura dei dati).

La variabile DL (Fig. 14)

La variabile DL è una chiave d'accesso al miniprogramma di recupero dati. Questo infatti viene eseguito soltanto quando DL=0. Con DL=1 il programma prosegue senza il recupero dei dati. Lo scopo di questa barriera d'accesso è quello di non richiedere il recupero dei dati da cassetta che una prima volta, quando si inizia il programma dal RUN, dopo di che, trovandosi i dati nella memoria RAM, si rende inutile caricarli di nuovo dal registratore, ciò anche in conseguenza del fatto che gestire dati da cassetta è disagiata per via delle operazioni manuali che si devono svolgere (riavvolgere il nastro e premere i tasti).

5400 CLOSE 1: DL=1

Dopo aver chiuso il file 1, la chiave DL si porta ad 1, in modo che i dati non verranno più recuperati dalla cassetta se non quando si riprenda dal RUN. Infatti, all'inizio del listato, troviamo la riga 5 DL=0, predisposta proprio a questa funzione. L'avvio del programma da RUN è necessario in 2 casi: quando il programma stesso sia appena stato caricato in memoria, oppure quando lo si chiuda premendo lo 0 dall'indice. Vedremo più avanti, quando verranno analizzati i tre sottoprogrammi, come ci si servirà della chiave DL per caricare i dati soltanto quando lo si ritiene necessario.

L'indice

È costituito dalla prima parte del programma n. 21 e va dalla riga 10 alla riga 190. Nella zona inferiore è stato inserito un INPUT di scelta con 5 numeri possibili. Rispondendo con 0 (zero), il programma si chiude. Fornendo invece in risposta un numero dall'1 al 4 (1 o 2 o 3 o 4), l'indice rimanda rispettivamente alle righe di programma n. 200 (per il n. 1) - n. 1000 (per il n. 2) - n. 2000 (per il n. 3) - n. 3000 (per il n. 4). L'espressione usata per la scelta è:

190 ON P GOTO 200, 1000, 2000, 3000

I numeri 200, 1000, 2000, 3000 sono stati scelti prima ancora di conoscere la lunghezza dei singoli programmi e quindi con abbondanza di righe libere tra un programma e l'altro. Complessivamente la distribuzione delle righe sarà:

10- 200 INDICE
200- 990 PROGRAMMA 1 - INTRODUZIONE DATI
1000-1990 PROGRAMMA 2 - CONSULTAZIONE DATI
2000-2990 PROGRAMMA 3 - AGGIORNAMENTO DATI
3000-3990 PROGRAMMA 4 - STAMPA DATI
5000-5410 SUBROUTINES.

Da notare, infine, la prima riga di programma: 5 DL=0. DL è una variabile che serve da chiave per il caricamento dei dati da cassetta. Di ciò si è già scritto nel corso di questa lezione.

Debugging

Effettuare il debugging significa spulciare il programma, cioè rivederlo in ogni singola sua parte per esaminarne il comportamento quando gli vengono fornite dall'esterno delle riposte intenzionalmente sbagliate. Lo scopo è quello di renderlo per quanto è possibile insensibile agli errori dell'operatore.

BASIC SCHOOL - Lezione n. 18

Problemi nell'osservazione del LISTING

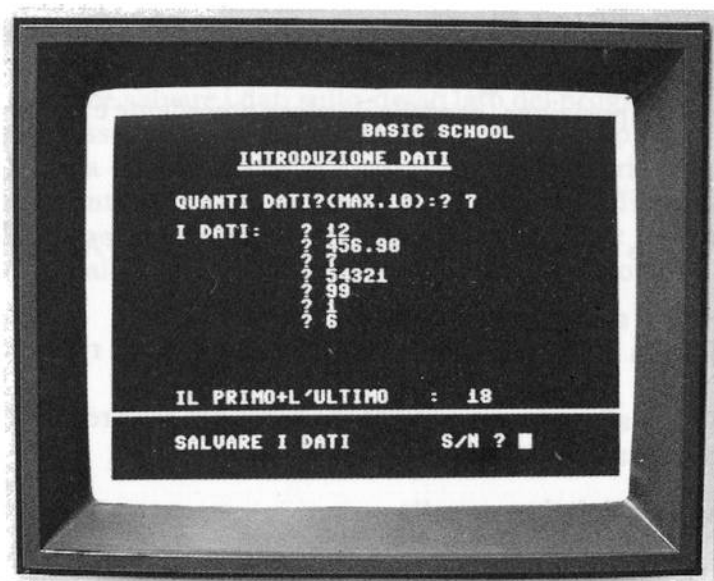
Quando le righe di programma diventano in numero superiore a quelle che può contenere lo schermo, si perde quella parte del programma che eccede la possibilità di visione dello schermo stesso.

Viene infatti proposta all'operatore soltanto la parte finale del programma. Per la scelta della sezione da osservare e come agire per visionare ciò che interessa, rimandiamo all'Appendice n. 1 dove vengono evidenziati i metodi per listare in maniera selettiva. Accenniamo qui ad un sistema per accedere più rapidamente alla parte di programma che interessa. Si scriva l'istruzione LIST, si batta il RETURN e quindi si tenga premuto il tasto CONTROL. Si vedrà il listato scorrere lentamente verso l'alto. Non appena si ha la certezza di essere sulla parte di listing che interessa esaminare, si preme il tasto RUN STOP.

Lezione pratica

Esercizio n. 21 - Seconda parte

Al sottoprogramma n. 1, chiamato "INTRODUZIONE DATI" si accede dall'indice premendo il n. 1 e il RETURN.



Esercizio n. 21
Sottoprogramma
di INTRODUZIONE
DATI, fermo al 1° INPUT.
Giunto alla fase
decisionale,
premendo S
comparirà la scritta
PRESS PLAY AND
RECORD ON TAPE (premi
i tasti PLAY e RECORD
sul registratore).
Premendo N, invece,
si ritornerà all'indice.

220 Alla chiave DL si dà il valore di 1 per non dover recuperare i dati da cassetta quando si passa internamente tra un quadro e l'altro.

Viene adesso creata la Maschera, con righe di programma che vanno da 250 a 320, quando viene scritta la frase "SALVARE I DATI S/N". Con le righe 330 e 340 viene riposizionato il cursore in modo da poter creare il primo INPUT, quello che riguarda la variabile N (numero dei dati da introdurre). Viene anche cancellato un eventuale numero fornito in precedenza.

Nel numero di riga 350 si annulla la variabile N, per azzerare i valori rimasti precedentemente in memoria e poi si crea l'INPUT.

Dopo di che, si sposta in basso il cursore di una battuta (riga 360), e si introducono (riga 370) i dati D(I) col sistema del FOR...NEXT. Ad ogni ciclo, il cursore viene TAB(15) in modo da stampare i dati con 15 battute di tabulazione orizzontale.

A questo punto, viene effettuato il calcolo $PU = D(1) + D(N)$, cioè viene sommato il primo dato con l'ultimo.

Con la riga 375 si istruisce il calcolatore a tener conto soltanto del dato D(1), quando N sia uguale a 1. Senza questa riga, quando $N=1$, l'unico dato fornito è allo stesso tempo il primo e l'ultimo e viene conteggiato due volte nella operazione della riga 380.

Adesso, si posiziona di nuovo il cursore al di là della frase (IL PRIMO + L'ULTIMO) e si scrive il valore PU.

Siccome non è possibile sapere a priori la posizione del cursore al termine dell'INPUT D(I), perchè il numero dei dati non è fisso, bisogna riportare il cursore stesso in alto a sinistra (posizione ottenibile col tasto HOME) e quindi scendere verso il basso di quanto occorra.

Ciò viene effettuato con le righe di programma

```
390 PRINT "S" (S è ottenibile premendo, dopo il 1° apice, il tasto HOME)
```

```
400 S = 18 : GOSUB 5010 (spazia 18 in verticale)
```

```
410 T = 30 : GOSUB 5070 : PRINT PU
```

N.B. Con la riga 358, qualora ad N venga attribuito un valore inferiore a 1 o superiore a 10, il cursore si riposiziona, cancella il valore precedente e ripete l'INPUT.

Scelta decisionale

420-430 Viene riposizionato il cursore al di là della frase "SALVARE I DATI S/N"

440 Viene creato un INPUT stringa per la variabile R\$.

450 - Se la risposta fornita è N, i dati non vengono salvati e si ritorna all'indice (riga 10)

460 - Se la risposta è S si va alla subroutine 5200 che serve per salvare i dati su cassetta, quindi si ritorna automaticamente all'indice.

Subroutine per salvataggio dei dati su cassetta

Spiegazione:

Con la 5240 si apre il file n. 1 (successivi files devono avere numeri diversi fino a 256) sulla cassetta (codice 1) per la scrittura (codice 1).

La sigla è "DATI"

Con la 5250 viene scritto su cassetta (PRINT# 1), cioè sul file 1, il valore della variabile N.

Con la 5260 vengono scritti su cassetta i valori di D(I) letti col sistema ciclico del FOR...NEXT.

Con la 5270 viene inciso su cassetta sul file 1 il valore della variabile PU.

Con la 5280 viene chiuso il file n. 1 prima aperto.

Giunti alla riga 5200, sullo schermo comparirà la frase PRESS PLAY AND RECORD ON TAPE, occorrerà premere i tasti PLAY e RECORD sul registratore e attendere che sullo schermo ricompaia il READY. A questo punto i dati sono registrati ed occorre riavvolgere subito il nastro fino al punto d'inizio di questa registrazione.

Ritorno automatico all'indice

Con la riga 460 che rimanda all'inizio del programma (riga 10), una volta riversati i dati sul nastro, il programma ritorna automaticamente all'indice, per le scelte operative del caso.

Gestione cassette

Per salvare i dati e quindi recuperarli in seguito, ci si può servire di una o due cassette. Si consiglia, per praticità, di utilizzarne due, una per caricare il programma sul calcolatore e l'altra per i dati, registrandoli a partire dall'inizio del nastro. Ogni volta che si saranno salvati i dati, si riavvolgerà completamente il nastro ed essi potranno venire subito consultati, con il programma n. 2.

In via alternativa, si può utilizzare la stessa cassetta, da un lato per il programma e dall'altro per i dati, tuttavia si perde più tempo perchè si deve riavvolgere completamente il nastro passando da un lato all'altro. Si possono anche salvare i dati sullo stesso lato del programma, incidendoli di seguito ad esso. È una soluzione migliore di quella appena esposta, ma un po' più delicata del sistema delle 2 cassette, perchè nel riavvolgere la parte di nastro che contiene i dati, si potrebbe oltrepassare di molto l'inizio degli stessi. Starà poi all'abilità di ogni operatore scegliere la soluzione che ritiene più vicina alle sue possibilità di operare con precisione.

Lezione pratica

Esercizio n. 21 - Terza parte

Al sottoprogramma n. 2, chiamato CONSULTAZIONE DATI, si accede dall'INDICE.

Alla domanda: SCEGLIERE IL PROGRAMMA, si dovrà rispondere con il 2 e battendo il RETURN.

Immediatamente comparirà sullo schermo la scritta PRESS PLAY ON TAPE, cioè PREMI IL TASTO PLAY SUL REGISTRATORE.

Occorre che preventivamente il nastro che contiene i DATI sia stato riavvolto in modo da poter ricominciare dall'inizio degli stessi.

La riga di programma 1020 provvede con l'istruzione IF DL = 1 THEN GOSUB 5300 al recupero dei dati dal nastro e il loro collocamento nella memoria RAM del calcolatore. Ciò avviene quando si parte dal RUN. Negli altri casi, la chiave DL=1 e risulta valida la riga 1030 IF DL=1 THEN GOTO 1070 e i dati non vengono recuperati dalla cassetta perchè sono già nella memoria del calcolatore.

Subroutine per caricamento dati da nastro

```
5360 OPEN 1,1,0, "DATI"  
5370 INPUT# 1, N (INPUT# = ingresso da periferica)  
5380 FOR I = 1 TO N : INPUT# 1, D(I) : NEXT I  
      (lettura dei dati della matrice avente per variabile D(I))  
5390 INPUT# 1, PU  
5400 CLOSE 1 (chiusura del file)  
5405 DL=1:REM NON RECUPERARE PIU' I DATI
```

Con la riga 5405, una volta recuperati i dati da cassetta, la chiave DL che prima era 0 (zero), diventa 1 e fintanto che non si uscirà dal programma, l'operazione di recupero non verrà più effettuata perchè inutile.

Maschera n. 2

Una volta incamerati, se necessario, in memoria RAM i dati recuperati dalla cassetta, occorre creare la maschera, scrivendo il titolo (1070-1110) e la parte di maschera riguardante la gestione dei dati con la GOSUB 5090.

Inserimento dati sullo schermo

A questo punto, i dati relativi alle variabili N, D(I) e PU recuperati dalla cassetta e memorizzati nella RAM, dovranno venir posizionati negli stessi punti che occupavano quando sono stati salvati al termine del programma "INTRODUZIONE DATI". Per raggiungere questo traguardo, occorre ri-

portare in alto e a destra il cursore così come si è fatto nelle righe 330-340 del programma 1.

1150 R=17 : GOSUB 5050 (per ritornare in alto)

1116 T=27 : GOSUB 5070 : PRINT N (per tabulare a destra e stampare il valore di N.

La stessa cosa avverrà per le altre variabili D(I) e PU. Per D(I) si dovranno leggere i dati indicizzati da 1 a N, con le seguenti formule:

1170 FOR I = 1 TO N : PRINT TAB(16) D(I) : NEXT I

I dati verranno così stampati sullo schermo uno sotto l'altro, con 16 battute di tabulazione.

Per il dato PU, la riga di programma che lo conduce a comparire sullo schermo a fianco della frase "IL PRIMO + L'ULTIMO" è:

1190 PRINT "☐" (per mandare il cursore in posizione HOME)

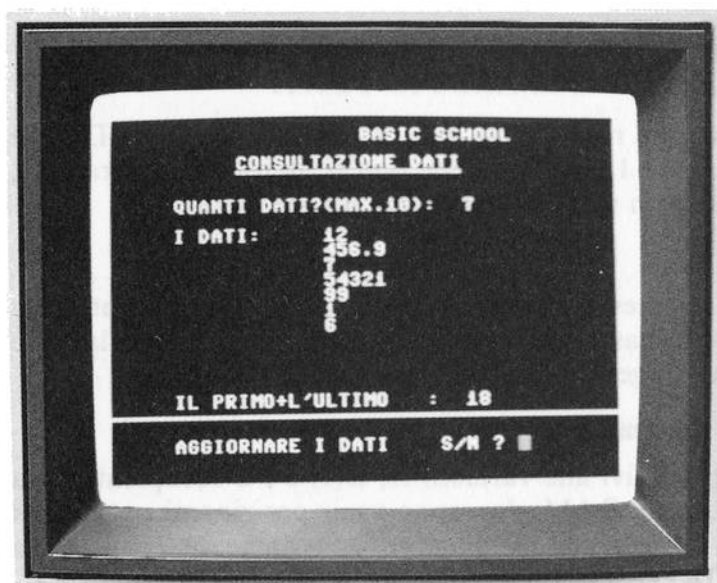
1200 S=18 : GOSUB 5010 (per far scendere il cursore di 18 battute)

1210 T=27 : GOSUB 5070 : PRINT PU (per tabulare di 27 battute e stampare il valore di PU.

1220-1230 Posizionamento del cursore a destra della frase AGGIORNARE I DATI S/N. Creazione di un INPUT per la variabile stringa RS.

1250-1260 Condizioni per ritornare all'indice oppure proseguire alla riga 2005 con il programma AGGIORNAMENTO DATI.

Come si può notare, qui non è necessario servirsi della chiave DL=1, perchè questa informazione è già presente nella memoria RAM, fornita o dalla riga 5405 qualora i dati siano recuperati da cassetta oppure dalla riga 220 se invece si è passati attraverso il sottoprogramma n. 1 INTRODUZIONE DATI.



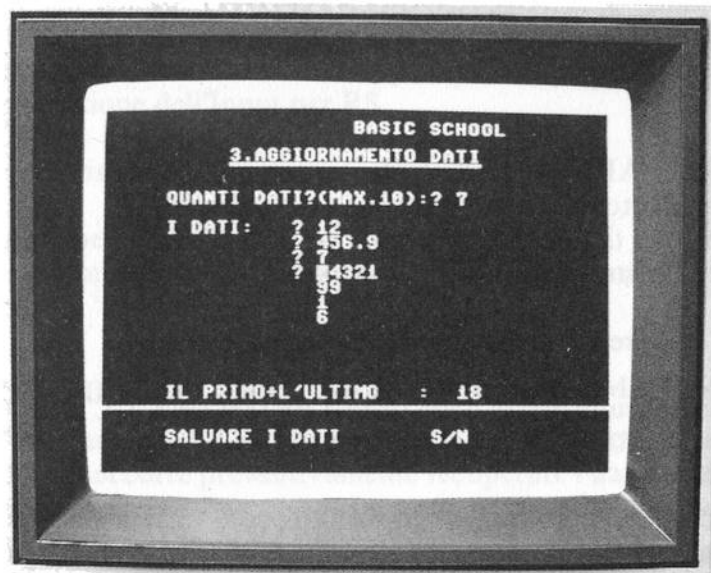
Esercizio n. 21
Sottoprogramma
di CONSULTAZIONE
DATI. Se vi si accede
dal RUN, occorre
preventivamente
caricare
i dati dalla cassetta.
Alla fase decisionale,
occorre premere
il tasto S se si intende
cambiare parzialmente
o totalmente
i dati consultati.

BASIC SCHOOL - Lezione n. 20

Esercitazione pratica

Esercizio n. 21 - Quarta parte

Al programma n. 3 intitolato AGGIORNAMENTO DATI, si può accedere dall' INDICE rispondendo con il n. 3 e RETURN alla domanda SCEGLIERE IL PROGRAMMA, oppure ci si può arrivare dal programma CONSULTAZIONE DATI rispondendo con una S alla domanda: AGGIORNARE I DATI S/N.



Esercizio n. 21
Sottoprogramma
di AGGIORNAMENTO
DATI. Il cursore
pulsante ritorna
progressivamente
sui dati già stampati,
che si possono
o meno sostituire
con altri. Premendo
semplicemente
il RETURN, verranno
confermati i dati
esistenti. Al termine
vi è la possibilità
di salvare
la nuova serie di dati.

Nel primo caso, la prima operazione che il calcolatore svolge è quella di caricare i dati dalla cassetta poichè l'aggiornamento viene effettuato cambiando i dati forniti in precedenza. Nel secondo caso, ciò non è più necessario perchè essi sono stati recuperati col programma di consultazione.

Sviluppo del programma

In pratica, si tratta di lavorare sul programma n. 2 di consultazione dati, ritornando con il cursore sui dati forniti in precedenza e cambiandoli quando occorra. I dati nuovi vengono sovrascritti su quelli vecchi che vengono in tal modo cancellati e sostituiti. La semplice operazione di battere il RETURN, lascia in memoria il dato precedente, cioè non apporta alcuna modifica.

Così come è facile l'enunciato, tanto meno lo è la realizzazione pratica del programma in quanto nascono diversi problemi di non facile risoluzione, come vedremo durante l'analisi dettagliata del listing.

I dati che si possono aggiornare dall'esterno sono: il valore di N, cioè del

numero di dati da inserire, e i singoli valori dei dati D(I). Il cambiamento di N con un valore inferiore rispetto al precedente comporta la cancellazione dei dati D(I) in eccesso. Il valore di PU, viene cambiato automaticamente, trattandosi di una operazione interna al calcolatore.

Analisi del listing

2000

Rinvio alla subroutine della riga 5300 per il recupero dei dati da cassetta quando DL=0, cioè quando, si parte dal RUN e si accede al programma 3 direttamente.

2005 - 2040

Titolazione del programma "3. AGGIORNAMENTO DATI"

2060

Rinvio alla subroutine della riga 5090 con la quale viene creata la maschera di base.

2070

Viene scritta la frase SALVARE I DATI S/N

2080 - 2090 Riposizionamento del cursore e stampa del valore di N.

2110

Scrittura dei dati D(I) col sistema della lettura ciclica FOR...NEXT.

2120

Riposizionamento del cursore in posizione HOME (alto a sinistra).

2130 - 2140

Si porta il cursore in basso, sulla destra della frase il PRIMO + L'ULTIMO e poi si stampa il valore di PU.

2150 - 2155

Si riporta il cursore sopra il numero che è stato fornito per la variabile N, e si crea un INPUT. Il cursore lampeggerà sopra la 1° cifra fornita in precedenza per N.

2165

Se ad N viene fornito un numero inferiore a 1 o superiore a 10, viene riposizionato il cursore, cancellato il numero sbagliato e rifatto l'INPUT.

2210 - 2230

Vengono ripetuti gli INPUT dei dati D(I) col sistema del FOR...NEXT, tabulando di 15 battute per ogni dato fornito senza azzerare i dati D(I).

2233 - 2239

Serve per cancellare i valori D(I) che sarebbero in eccesso qualora venisse fornito un nuovo valore di N inferiore a quello precedente.

Esempio: Se il vecchio valore di N era 7 ed il nuovo è 4, il nuovo valore di I all'uscita del FOR...NEXT è 5.

Perciò, i dati dal 5° al 10° vengono cancellati con la riga 2233.

2240

Riposizionamento del cursore in posizione HOME.

2250

Abbassamento del cursore per 18 battute.

2255

Se viene fornito un solo dato, cioè se $N=1$, $PU=D(1)$ e si salta la riga 2260.

2260

Calcolo del nuovo valore di $PU= D(1)+D(N)$.

2270

Tabulazione del cursore per portarlo sulla destra della frase IL PRIMO + L'ULTIMO.

2275

Stampa del valore di PU. Il cursore viene portato dapprima tutto a destra con la barra spaziatrice, poi viene fatto ritornare verso sinistra con il CRSR (comandato dall'interno) fino ad assumere la posizione giusta. Ciò per cancellare il valore di PU scritto in precedenza che può essere più lungo di quello attuale.

2280 - 2290

Posizionamento del cursore a destra della frase SALVARE I DATI S/N e creazione dell'Input per R\$.

2310

Se la risposta è N, si ritorna all'indice.

2320

Altrimenti, cioè se la risposta è S, si opera il salvataggio dei dati con la subroutine 5200 e poi si ritorna automaticamente all'indice.

STAMPA DEI DATI - SOTTOPROGRAMMA N. 4

Vi si accede dall'indice premendo 4 e il tasto Return. Se si proviene dal RUN, occorre preventivamente recuperare i dati dalla cassetta, altrimenti la macchina non saprebbe che cosa stampare. La riga che comanda questa operazione è la 3010. Provenendo dall'interno, $DL=1$ e siccome i dati sono già in memoria, il recupero è annullato (riga 3020).

3010 Se $DL=0$, recupera i dati da cassetta (sub. 5200)

3020 Se $DL=1$ prosegui nel programma

3030 Apri il file n. 1 per la stampante (codice 4): predisponi la stampante.

3040 Ripeti 80 volte il trattino per fare una linea di separazione

3110 Scrivi (tab.5) la frase QUANTI DATI SONO STATI FORNITI: e accanto il valore di N.

3120 Scrivi (tab.5) la frase DESCRIZIONE DEI DATI e accanto il valore del 1° dato $D(1)$.

3130 Per I che va da 2 a N, scrivi (tab. 40) il valore dei dati $D(I)$ partendo dal 2° dato.

Si è trattato di un artificio per avere il primo dato in linea con la frase della riga 3120.

3150 Scrivi (tab. 5) la frase SOMMA DEL PRIMO DATO + L'ULTIMO e, accanto, il valore di PU.

3170 Fai una riga di chiusura.

3180 Stacca la stampante, chiudi il file e ritorna all'indice. Un esempio di stampa lo abbiamo alla fig. 15.

In figura 14 è rappresentato un quadro riassuntivo del funzionamento a blocchi del programma di gestione dati con indicazione della chiave DL per il recupero dei dati da cassetta quando si accede al sottoprogramma direttamente dal RUN. Come si può notare, partendo dal RUN, DL=0. Premendo

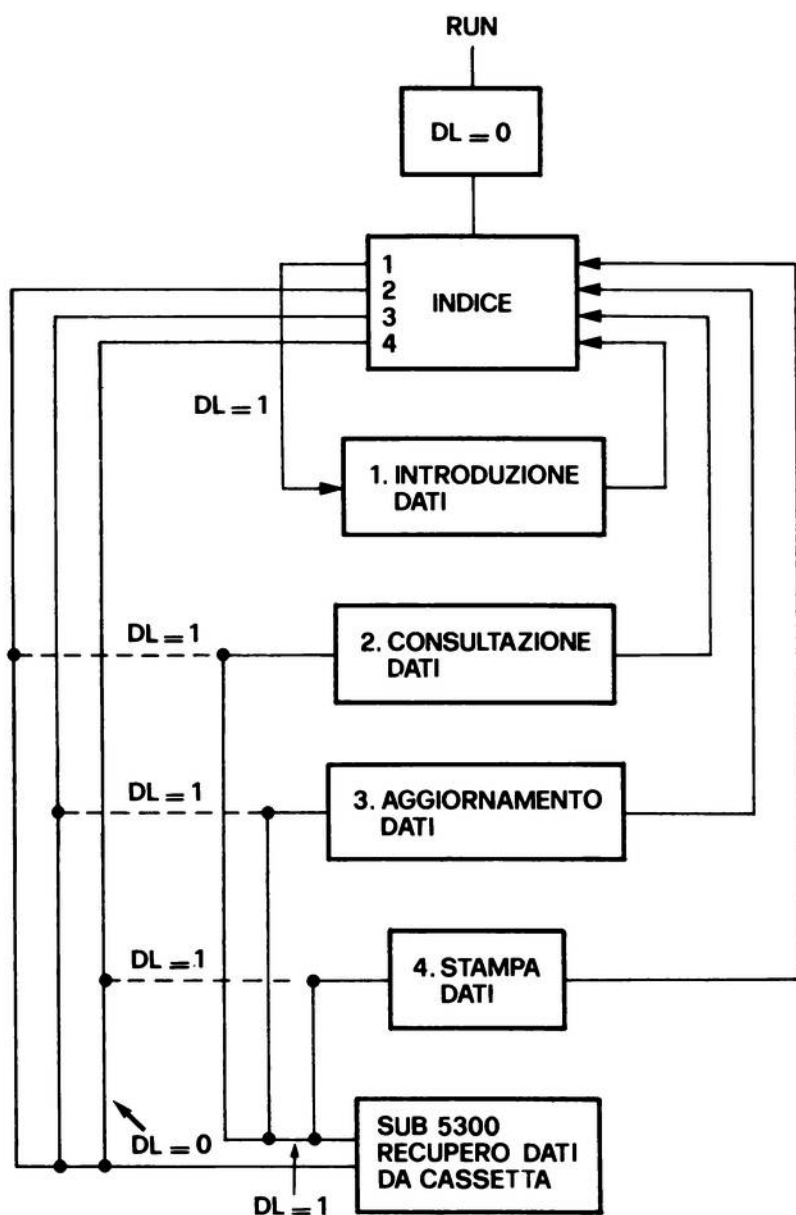


Fig. 14

subito 2, 3 oppure 4, il DL=0 e porta alla subroutine 5300 di recupero dei dati. Dopo questa operazione DL=1 e anche all'ingresso dell'INTRODUZIONE DATI DL=1 cosicchè, quando si accede ai quadri 2, 3 e 4 dall'interno, il recupero non avviene più perchè inutile (i dati sono nella memoria RAM).

PROVA PRATICA DI STAMPA DATI

BASIC SCHOOL 21

STAMPA DEI DATI

| | |
|----------------------------------|----------|
| QUANTI DATI SONO STATI FORNITI | 10 |
| DESCRIZIONE DEI DATI | 234 |
| | 23.98 |
| | 678 |
| | 4321567 |
| | 67.45678 |
| | 34 |
| | 2 |
| | 1987654 |
| | 34 |
| | 8 |
| SOMMA DEL PRIMO DATO + L'ULTIMO: | 242 |

Fig. 15 - Prova di stampa ottenuta fornendo 10 dati nel programma INTRODUZIONE DATI e passando in seguito al sottoprogramma 4 intitolato STAMPA DATI. Per ragioni di spazio, i numeri sono resi più compatti di quanto non lo siano nell'esecuzione reale. Inoltre, l'intestazione BASIC SCHOOL 21 è stata spostata sulla sinistra, mentre nella realtà essa è posizionata sulla destra del foglio.

LISTATO ESERCIZIO 21

```
5 DL=0
10 PRINT"3"
20 PRINT,,"BASIC SCHOOL 21"
30 SP=2:GOSUB5010
40 PRINT TAB(8)"INDICE PER GESTIONE DATI"
45 PRINT TAB(8)"-----"
50 SP=2:GOSUB5010
```

```

60 PRINT TAB(10)"1.INTRODUZIONE  DATI"
70 PRINT
80 PRINT TAB(10)"2.CONSULTAZIONE DATI"
90 PRINT
100 PRINT TAB(10)"3.AGGIORNAMENTO DATI"
103 PRINT
106 PRINT TAB(10)"4.STAMPA DATI"
110 SP=2:GOSUB5010
120 PRINT TAB(10)"0.CHIUSURA PROGRAMMA"
130 PRINT
140 GOSUB5030
150 SP=2:GOSUB5010
160 INPUT"■■■■■■■■SCEGLIERE IL PROGRAMMA";P
170 IF P=0 THEN END
190 ON P GOTO 200,1000,2000,3000
200 REM
210 REM INTRODUZIONE DATI
220 DL=1: REM NON RECUPERARE I DATI
250 PRINT"J"
260 PRINT,,"BASIC SCHOOL"
270 PRINT
280 PRINT TAB(10)"INTRODUZIONE DATI"
290 PRINT TAB(10)"-----"
310 GOSUB5090
320 PRINT TAB(5)"SALVARE I DATI          S/N"
330 R=17:GOSUB5050
340 T=26:GOSUB5070:PRINT"      ■■■■■■";
350 N=0:INPUT N
358 IF N<1 OR N>10 THEN PRINT CHR$(145);:
      GOTO 340
360 PRINT
370 FOR I=1 TO N:T=15:GOSUB 5070:D(I)=0:INPUT
      D(I):NEXT I
375 IF N=1 THEN PU=D(1):GOTO 390
380 PU=D(1)+D(N)
390 PRINT"Q"
400 SP=18:GOSUB5010
410 T=27:GOSUB5070:PRINT PU
420 PRINT
430 T=30:GOSUB5070
440 INPUT R$

```

```

450 IF R$="N" THEN GOTO 10
460 IF R$="S" THEN GOSUB 5200:GOTO 10
470 END
990 REM
1000 REM CONSULTAZIONE DATI
1010 REM
1020 IF DL=0 THEN GOSUB 5300:REM RECUPERA
    I DATI
1030 IF DL=1 THEN GOTO 1070:REM NON
    RECUPERARE I DATI
1070 PRINT"3"
1080 PRINT,,"BASIC SCHOOL"
1090 PRINT
1100 PRINT TAB(10)"CONSULTAZIONE DATI"
1110 PRINT TAB(10)"-----"
1120 GOSUB 5090
1140 PRINT TAB(5)"AGGIORNARE I DATI    S/N"
1150 R=17:GOSUB 5050
1160 T=27:GOSUB 5070:PRINT N
1165 PRINT
1170 FOR I=1TON:PRINT TAB(16)D(I):NEXT I
1190 PRINT"4"
1200 SP=18:GOSUB 5010
1210 T=27:GOSUB 5070:PRINT PU
1220 PRINT
1230 T=30:GOSUB 5070:INPUT R$
1250 IF R$="N" THEN GOTO 10
1260 IF R$="S" THEN GOTO 2005
1980 REM
1985 REM AGGIORNAMENTO DATI
1990 REM
2000 IF DL=0 THEN GOSUB 5300: REM RECUPERA
    I DATI
2003 IF DL=1 THEN GOTO 2005:NON RECUPERARE
    I DATI
2005 PRINT"3"
2010 PRINT,,"BASIC SCHOOL"
2020 PRINT
2030 PRINT TAB(10)"3.AGGIORNAMENTO DATI"
2040 PRINT TAB(10)"-----"
2060 GOSUB 5090

```

```

2070 PRINT TAB(5)"SALVARE I DATI           S/N"
2080 R=17:GOSUB5050
2090 T=27:GOSUB5070:PRINT N
2100 PRINT
2110 FOR I=1TON:PRINT TAB(16)D(I):NEXT I
2120 PRINT"☺"
2130 SP=18:GOSUB5010
2140 T=27:GOSUB5070:PRINT PU
2150 R=14:GOSUB5050
2155 T=26:GOSUB5070
2160 INPUT N
2165 IF N<1 OR N>10 THEN PRINTCHR$(145);:
      PRINTTAB(26)"          ██████████";GOTO 2160

2200 PRINT
2210 FOR I= 1 TO N
2220 T=15:GOSUB5070:INPUT D(I)
2230 NEXT I
2233 FORK=IT010
2236 PRINT TAB(16)"          "
2239 NEXT K
2240 PRINT"☺"
2250 SP=18:GOSUB5010
2255 IF N=1 THEN PU=D(1):GOTO 2270
2260 PU=D(1)+D(N)
2270 T=27:GOSUB5070
2275 PRINT"          ██████████" PU
2280 PRINT
2290 T=30:GOSUB5070:INPUT R$
2310 IF R$="N"THEN GOTO 10
2320 IF R$="S"THEN GOSUB 5200:GOTO 10
2330 END
3000 REM STAMPA DATI
3010 IF DL=0THEN GOSUB 5300
3020 IF DL=1 THEN 3030
3030 OPEN1,4 : CMD1
3040 FOR K=1 TO 80: PRINT"~"; NEXT K
3050 PRINT:PRINT
3060 PRINT TAB(50)"BASIC SCHOOL 21"
3070 S=3 : GOSUB 5010
3080 PRINT TAB(30)"STAMPA DEI DATI"

```

```

3090 PRINT TAB(30)"-----"
3100 PRINT
3110 PRINT TAB(5)"QUANTI DATI SONO STATI
    FORNITI : "N
3115 PRINT
3120 PRINT TAB(5)"DESCRIZIONE DEI
    DATI : "D(1)
3130 FOR I= 2 TO N:PRINT TAB(40) D(I): NEXT I
3140 PRINT
3150 PRINT TAB(5)"SOMMA DEL PRIMO DATO +
    L'ULTIMO: "PU
3160 PRINT:PRINT
3170 FOR K=1 TO 80: PRINT"~"; NEXT K
3180 PRINT#1: CLOSE1:GOTO 10
5000 END
5005 REM SUB CHE SPAZIA VERSO IL BASSO
5007 REM
5010 FOR K=1TO SP :PRINT:NEXTK:RETURN
5020 REM
5025 REM SUB PER LINEA DI SEPARAZIONE
5027 REM
5030 FORK=1TO40:PRINT"-":NEXTK:RETURN
5040 REM
5045 REM SUB PER RITORNARE IN ALTO
5047 REM
5050 FORK=1TO R:PRINT CHR$(145):NEXTK:RETURN
5060 REM
5065 REM SUB PER TABULARE
5067 REM
5070 PRINT TAB(T):RETURN
5080 REM
5085 REM SUB PER MASCHERA
5087 REM
5090 PRINT
5095 PRINTTAB(5)"QUANTI DATI?(MAX.10):"
5100 PRINT
5110 PRINT TAB(5)"I DATI:"
5120 FORK=1TO10:PRINT:NEXTK
5130 PRINT TAB(5)"IL PRIMO+L'ULTIMO : "
5140 GOSUB5030

```

```
5150 PRINT
5160 RETURN
5200 REM
5210 REM SALVATAGGIO DATI SU CASSETTA
5230 REM
5235 PRINT
5240 OPEN1,1,1,"DAT1"
5250 PRINT# 1,N
5260 FOR I=1TON:PRINT# 1,D(I):NEXTI
5270 PRINT# 1,PU
5280 CLOSE1
5290 RETURN
5300 REM
5310 REM RECUPERO DATI DA CASSETTA
5320 REM
5360 OPEN1,1,0,"DAT1"
5370 INPUT# 1,N
5380 FOR I=1TON:INPUT# 1,D(I):NEXTI
5390 INPUT# 1,PU
5400 CLOSE1
5405 DL=1: REM NON RECUPERARE PIU' I DATI
5410 RETURN
```

Riassunto delle funzioni INPUT - RUN - LIST - PRINT

INPUT

INPUT L (oppure INPUT LU)

Forma un punto di domanda al quale occorre rispondere con un dato numerico intero o con virgola flottante. Non impiegare sigle che si possano confondere con le istruzioni BASIC (es.: TO si confonde con il TO di FOR K= 1 TO ecc.).

INPUT L% (oppure INPUT LU%)

Fornire soltanto un dato numerico che sia intero.

INPUT L\$ (oppure LU\$)

Fornire un dato stringa (solo lettere oppure numeri e lettere assieme).

INPUT "LUNGHEZZA =" ; L

Scrivi sullo schermo la frase LUNGHEZZA= ?

Al punto di domanda, fornire la risposta (in questo caso si tratta di un numero). Per la stampa della frase che è scritta fra gli apici, vedere quanto viene detto per la funzione PRINT.

Una volta aperto il primo apice, prima di scrivere la frase esplicativa, potranno essere printate delle funzioni CHR\$ come ad esempio i segni grafici che indicano spostamenti del cursore (oppure).

RUN

Esegui l'intero programma. Se esistono dei punti di domanda (INPUT) occorre fornire i dati relativi, se no l'esecuzione si arresta. Per uscire rapidamente dal programma premere RUN/STOP e RESTORE.

RUN 400

Esegui il programma a partire dalla riga 400.

LIST

Lista tutto il programma mostrando la parte finale quando esso sia troppo lungo per comparire intero sullo schermo.

LIST 100

Mostra soltanto la riga 100.

LIST -100

Mostra il programma fino alla riga n. 100

LIST 150 — 300

Mostra le righe di programma comprese fra 150 e 300.

LIST 400 —

Mostra il listing a partire dalla riga 400.

PRINT

Il cursore si sposta da sinistra a destra senza scrivere niente e ritorna di nuovo a sinistra spostato di una battuta più in basso rispetto alla posizione che occupava alla partenza. Serve per fornire uno spazio in verticale.

PRINT:PRINT:PRINT, ecc. (abbrev. ? : ? : ? :)

Più spazi in verticale. Di preferenza si scrivono sulla stessa linea.

PRINT Y (Y%)

Stampa il valore numerico della variabile rappresentata con la lettera Y.

PRINT Y X Z

Stampa i valori, uno accanto all'altro delle tre variabili Y,X,Z.

PRINT Y ;

PRINT "FRASE"

Stampa il valore di Y e a fianco la (1) "FRASE".

PRINT "FRASE"

Stampa la frase compresa fra gli apici.

PRINT, "FRASE"

Stampa la frase con tabulazione automatica di 10 battute dal margine sinistro.

PRINT TAB(8) "FRASE"

Stampa la frase iniziando da 8 spazi a partire dal margine sinistro. Attenzione, TAB(8) va tutto attaccato.

PRINT TAB(8) "FRASE 1" TAB(20) "FRASE 2"

Scrivi tabulando 8 la FRASE 1 e tabulando 20 la FRASE 2.

PRINT TAB(8) "FRASE 1" SPC(10) "FRASE 2"

Scrivi tabulando 8 la FRASE 1 e spaziando di 10 battute dalla frase 1 la FRASE 2.

PRINT CHR\$(145);

Sposta il cursore in alto di uno spazio.

Da qui deriva che PRINT non significa in questo caso (e in altri simili)

STAMPA, ma ESEGUI la funzione CHR\$(145).

PRINT "☒"

Significa PRINT "SHIFT—CLR/HOME" ed ha la stessa funzione di ripulire lo schermo e di mandare il cursore in alto a sinistra. Es.: quando si effettua il RUN di un programma.

PRINT " "

Serve per ottenere uno spazio a piacere (basta spostare l'apice di destra).

PRINT SQR(X)

Serve per rendere esecutive e stampare le diverse funzioni.

Qui $SQR(X) = \sqrt{X}$.

PRINT TAB(X)

Sposta il cursore a destra di x battute.

Salvataggio dati su cassetta

```

n. file ——— |
              | =cassetta
              |
OPEN 1,1,1, "SIGLA DATI"
              |
              | scrittura
              |
PRINT# 1, V   |
              | 1 spazio
PRINT# 1, U   |
              | 1 spazio
              |
              | V e U sono variabili dei dati da salvare
              |
CLOSE 1
    
```

N.B. In uno stesso programma possono essere aperti più file contemporaneamente che andranno numerati in modo progressivo.

Salvataggio dati su disco

```

8 = floppy disk ——— |
                    |
OPEN 1,8,1, "SIGLA DATI"
PRINT# 1, V
                    |
                    | 1 spazio
PRINT# 1, U
                    |
                    | 1 spazio
FOR I = 1 TO N : PRINT# 1, D(I) : NEXT I
CLOSE 1
N.B. Il FOR...NEXT si impiega quando si devono leggere dati
indicizzati con matrici.
    
```

Recupero dati da cassetta

```

n. file ——— |
             | cassetta
             |
OPEN 1,1,0, "SIGLA DATI"
INPUT# 1, V
             |
             | lettura
             |
INPUT# 1, U
             |
             | 1 spazio
             |
             | 1 spazio
CLOSE 1
    
```

Recupero dati da disco

8 = floppy disk —┐
OPEN 1,8,0, "SIGLA DATI"
INPUT#1, V
└─┬─ 1 spazio
INPUT#1, U
└─┬─ 1 spazio
FOR I = 1 TO N : INPUT# 1, D(I) : NEXT I
CLOSE 1

Stampa listato su carta

Il codice della stampante è 4. Tra un comando e l'altro battere il tasto RETURN

stampante —┐
OPEN 1,4
CMD 1
LIST (attendere la stampa del listato)
PRINT#1 (tutto attaccato)
CLOSE 1

Stampa esecuzione programma su carta

```
10 OPEN 1,4
20 CMD 1
30 PRINT "FRASE 1"
40 PRINT "FRASE 2", ecc. ecc.
50 PRINT V (variabile V)
60 PRINT U (variabile U)
70 FOR I = 1 TO N : PRINT D(I) : NEXT I
80 CLOSE 1
```

Per determinare i valori dispari (o pari) di una serie di dati

Sia N il numero dei dati forniti:

```
FOR I = 1 TO N
R = D(I) — (INT(D(I) / 2)* 2) (INT = Integer = Intero del numero)
IF R = 1 THEN PRINT D(I), "È DISPARI"
NEXT I
```

Riordino dati

Esempio: debbansi riordinare per ordine crescente i seguenti 5 dati

5 1 3 2 4
D(1) D(2) D(3) D(4) D(5)

Occorre che ogni valore di $D(I)$ venga confrontato con il valore di tutti i dati che seguono $D(J)$. Quando capita che $D(I)$ sia superiore di $D(J)$, bisogna che i due valori vengano scambiati fra di loro. (Fig. 16).

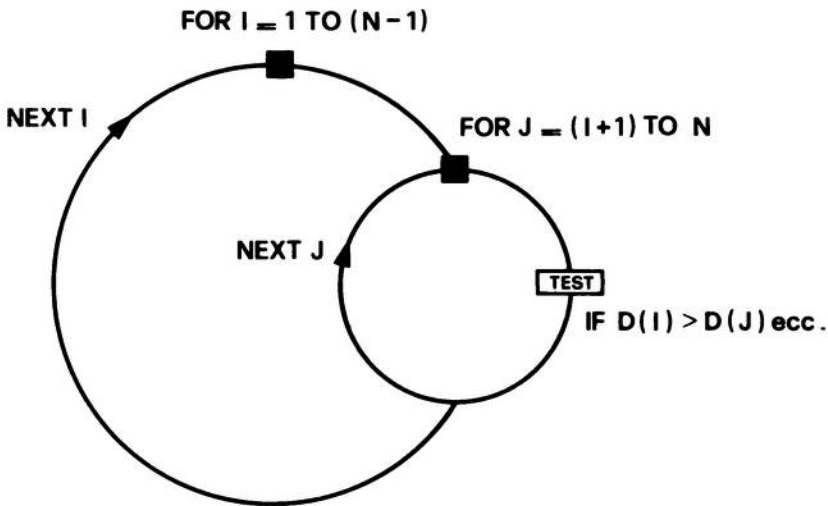


Fig. 16

```
10 FOR I = 1 TO (N - 1)
20 FOR J = (I + 1) TO N
30 IF D(I) <= D(J) THEN 70
40 T = D(I)
50 D(I) = D(J) scambio dei due valori
60 D(J) = T
70 NEXT J
80 NEXT I
90 FOR I=1 TO N : PRINT D(I) ; : NEXT I
```

Letture programma di fig. 16

10 per I da 1 a $(N-1)$.

Ogni valore $D(I)$ va confrontato con tutti gli altri ma non con se stesso. Nel caso presente $N=5$ ma il confronto va fatto 4 volte.

20 per J da (I+1) a N

J è il dato che segue I, quindi inizia con (I+1) per arrivare all'ultimo dato N.

30 Se il 1° dato cioè D(I) è superiore al 2°, cioè D(J). Esegui scambio.

60 È il programma di scambio dei dati che avviene soltanto se il dato D(I) è superiore al dato D(J) (riga 30).

In questo caso la formula è:

40 $T = D(I)$.

T è una variabile temporanea per parcheggiare il dato D(I) da trasferire.

50 $D(I) = D(J)$.

Con questa istruzione, il dato D(J) viene trasferito nella casella D(I) rimasta vuota.

60 $D(J) = T$.

Con questa istruzione, il dato T che in precedenza era nella casella D(I) viene ora trasferito nella casella D(J) rimasta vuota dopo l'esecuzione della riga 50.

90 Stampa i dati D(I) così riordinati, uno a fianco dell'altro.

S=S+P (sommatoria)

Viene utilizzata quando si intende fare la sommatoria progressiva di più numeri. Occorre far leggere con un'operazione ciclica FOR...NEXT tutti i dati in successione progressiva (continua o a passo).

Ad ogni passaggio, la variabile S si incrementa del dato che segue (P). Così, nella formula sopra esposta, la prima S è la somma finale dopo aver aggiunto il valore di P, mentre la S posta dopo l'uguale è la somma calcolata al passaggio precedente. Nell'elettronica digitale è possibile ciò che nella normale matematica appare astruso, cioè che una variabile sia uguale a se stessa sommata ad un altro dato, cioè che $S = S+P$.

Le correzioni si distinguono in 2 specie:

1. Correggere mentre si forma una frase. Si ritorna indietro col cursore premendo il tasto **DEL** (Delete) cancellando quanto si debba rifare.
2. Si ritorna indietro col cursore senza cancellare (ad esempio quando si debba correggere una lettera oppure un numero). Si ribatte quanto sbagliato e si ritorna col cursore in avanti per riprendere l'enunciato. Per queste operazioni si impiegano i tasti **CRSR** con o senza **SHIFT** per spostare il cursore senza cancellare, mentre si usa la barra spaziatrice per cancellare la lettera sbagliata.
3. Correggere una riga di programma quando sia già stato battuto il **RETURN**, cioè sia già stata immagazzinata in memoria la riga di istruzione. Per i primi tempi è meglio rifare la riga utilizzando lo stesso numero di riferimento. Quando verrà battuto il **RETURN** la riga nuova sostituirà completamente la vecchia.
4. In seguito si imparerà a correggere quanto di sbagliato oppure ad aggiungere o a togliere quanto vada modificato, senza rifare l'intera riga di programma che a volte è assai lunga per poterla ricordare a memoria.
5. Esempio 1. Si vuole cambiare una lettera o un numero. Si va con **SHIFT/CRSR** sulla lettera o sul numero da cambiare, si batte il nuovo carattere e poi il **RETURN**. Quindi, sempre con **CRSR** si porta il cursore in fondo oltre il **READY** e si riprende il programma o si esegue il **RUN**.
6. Esempio 2. Si vogliono introdurre una o più lettere o numeri o segni nella riga già fatta. Si va col cursore sulla lettera o sul numero che sta a destra dello spazio che si vuole ottenere. Si premono **SHIFT** e **INST** e si crea lo spazio necessario per contenere le nuove battute che si effettueranno subito, quindi si batterà il **RETURN**.
7. Se diverse di queste operazioni si dovranno effettuare in una stessa riga di programma, conviene ad ogni correzione elementare premere il **RETURN** e poi riportare il cursore per la correzione a venire.
8. Se, al contrario, si vuole spostare a sinistra una parte della riga per annullare spazi inutili, comportarsi come sopra, ma premere soltanto **DEL**, senza lo **SHIFT**. Ricordarsi sempre che al termine delle correzioni occorre uscire in basso oltre il **READY** prima di proseguire.
9. Per cancellare una riga di programma, basta riscrivere il numero della riga da annullare e battere il **RETURN**.
10. Per cambiare una riga di programma, si porta il cursore sopra il numero di quella riga, si scrive il numero nuovo e si batte il **RETURN**. Quindi, si esce in basso oltre il **READY** e si cancella il numero vecchio col sistema indicato in 9.

APPENDICE 4 - Funzioni varie

Funzione FRE(X)

Quando si presume di aver consumato un notevole spazio di memoria, una volta caricato il programma ed eventuali DATI, battere

```
PRINT FRE(X)
```

Comparirà sullo schermo il numero dei bytes ancora liberi.

Funzione RND (1)

RND significa RANDOMIZE (casuale)

L'istruzione da fornire al calcolatore per la scelta casuale di numeri è la seguente

```
10 PRINT RND (1)
```

Essa fornisce un numero a caso da 0 a 1 senza toccare gli estremi.

Per superare l'1, essa va così modificata

```
10 PRINT RND (1)+1
```

(N.B.: Con questa formula sono esclusi sia l'1 che il 2).

Qualora si voglia un numero superiore all'unità, occorre moltiplicare l'espressione per il numero limite superiore. Ad esempio, per ricercare un numero casuale da 1 a 10, essa assume la seguente forma

```
10 PRINT (10 * RND(1))+1
```

Qualora poi si vogliano soltanto dei numeri interi, si userà la funzione INT (Integer-Intero) che di qualsiasi numero, sceglie soltanto quanto è a sinistra della virgola. Es.

```
10 PRINT INT (10 * RND(1))+1
```

Esempio pratico: Far stampare 5 numeri interi a caso tra 1 e 10.

```
10 FOR K = 1 TO 5  
20 PRINT INT (10 * RND(1))+1  
30 NEXT K
```

Se si vogliono cambiare entrambi i limiti occorre impiegare la formula:

```
INT ((Ls-Li)*RND(1)+Li)+1
```

dove Ls=Limite superiore e Li=Limite inferiore.

Per fornire più INPUT uno accanto all'altro

Può essere necessario per questioni di spazio, specialmente nel campo didattico, poter fornire INPUT a ripetizione, uno accanto all'altro e non distribuiti verticalmente come capita più normalmente. La formula da impiegare è:

```
10 FOR I = 1 TO N  
20 INPUT " [ ] [ ] [ ] [ ] "; D(I)  
30 PRINT " [●]";  
40 FOR L = 1 TO I * 5 : PRINT " [ ]" ; : NEXT L  
50 NEXT I
```

Dopo il primo INPUT, riga 20 e, una volta incamerato il dato fornito dall'operatore, il cursore si trova sul lato sinistro dello schermo e spostato una battuta più sotto. Con la riga 30, esso risale sulla riga precedente, e con la riga 40 si sposta verso destra di un numero di battute che è progressivamente crescente. (Fig. 17).

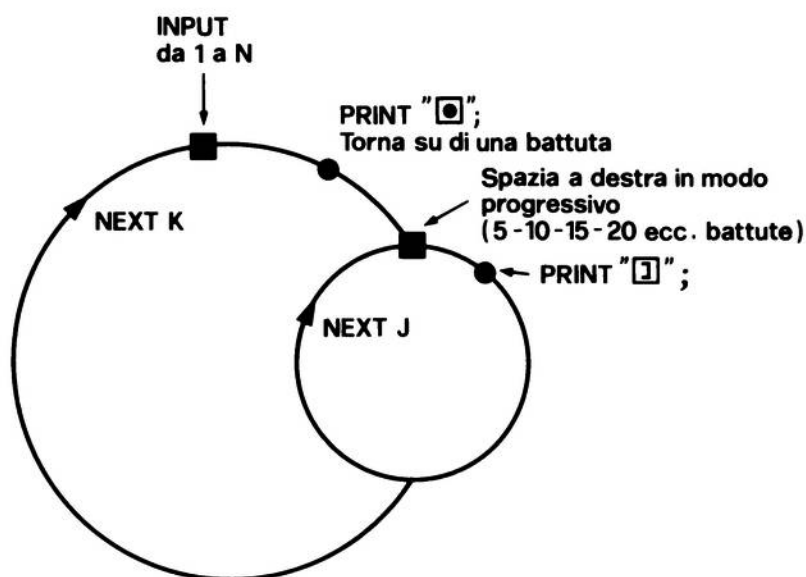


Fig. 17

Infatti, per $I = 1, L = 5$
 per $I = 2, L = 10$
 per $I = 3, L = 15$

Se ne deduce che il punto interrogativo e il cursore pulsante dell'INPUT vengono scritti in successione dopo 5,10,15 battute orizzontali più a destra del primo INPUT.

N determina il numero dei cicli di ripetizione delle istruzioni INPUT. Il moltiplicatore di I (riga 40), che qui è 5, determina le battute di distanza fra i diversi INPUT.

N.B. Questa formula è valida però soltanto fino all'esaurimento di una riga orizzontale. Per più dati occorre apportare correzioni che sono troppo complesse per questo primo corso di base.

APPENDICE 5 - Indice degli esercizi della cassetta

| SIGLA PER IL LOAD | ARGOMENTO DI MASSIMA | N. GIRI |
|-------------------------|--|-----------|
| Lato A | | |
| 1 | INDIRIZZO | 0 : 10 |
| 2 | INDIRIZZO - 1 ^a variazione | 12 : 19 |
| 3 | INDIRIZZO - 2 ^a variazione | 21 : 29 |
| 4 | CALCOLO AREA RETTANGOLO | 31 : 39 |
| 5 | CALCOLO AREA RETTANGOLO - 1 ^a variazione | 41 : 50 |
| 6 | CONDIZIONALE E SALTO CON RIPETIZIONE IF | 52 : 62 |
| 7 | SCELTA CONDIZIONATA | 64 : 73 |
| 8 | RIPETIZIONE COL GET - 1 ^a variazione | 75 : 85 |
| 9 | OPERAZIONE CICLICA PIU' SOMMATORIA | 87 : 95 |
| 10 | OPERAZIONE CICLICA A PASSO CON GOSUB | 97 : 105 |
| 11 | OPERAZIONE CICLICA INVERSA - 1 ^a variazione | 107 : 115 |
| 12 | DATARIO SETTIMANALE CON IF...THEN | 117 : 124 |
| 13 | DATARIO SETTIMANALE CON ON | 126 : 134 |
| 14 | DATARIO CON READ APERTO | 162 : 142 |
| 15 | DATARIO CON READ CHIUSO | 144 : 152 |
| Lato B | | |
| 16 | PROVA DI READ CON 10 NUMERI | 0 : 11 |
| 17 | OPERAZIONE CICLICA DOPPIA | 13 : 22 |
| 18 | MATRICE CON DATI INTERNI | 24 : 36 |
| 19 | MATRICE CON DATI ESTERNI | 38 : 47 |
| 20 | MATRICE CON MASCHERA | 50 : 59 |
| 21 | ESERCIZIO FINALE GESTIONE DATI | 63 : 82 |

LISTATO ESERCIZIO 1

```
10 PRINT,,"BASIC SCHOOL 1"  
20 PRINT:PRINT:PRINT  
30 PRINT,"ROSSI GIOVANNI"  
40 PRINT  
50 PRINT,"VIA CURTATONE 55"  
60 PRINT  
70 PRINT,"20052 MONZA"  
80 PRINT  
90 PRINT,"MILANO"
```

LETTURA ESERCIZIO 1

N.B. Facciamo notare che i caratteri grafici ottenuti con la stampante non sono identici a quelli che compaiono sullo schermo del monitor.

- 10 Scrivi la frase "BASIC SCHOOL 1" tabulando automatico 2 (PRINT,, "FRASE")
- 20 Spazia in verticale di 3 interlinee (I due punti (:)) separano le 3 istruzioni che sono scritte sulla stessa riga di programma)
- 30 Scrivi "ROSSI GIOVANNI" tabulando automatico 1 (PRINT, "FRASE")
- 40 Spazia in verticale di 1 interlinea
- 50 Ecc. ecc.

LISTATO ESERCIZIO 2

```
10 PRINT,,"BASIC SCHOOL 2"  
20 PRINT:PRINT:PRINT  
30 PRINT,"ROSSI GIOVANNI"  
40 PRINT:PRINT  
50 PRINT,"VIA CURTATONE 55"  
60 PRINT  
70 PRINT,"20052 MONZA"  
80 PRINT  
90 PRINT,,"MILANO"
```

LETTURA ESERCIZIO 2

- 10 Scrivi la frase "BASIC SCHOOL 1" tabulando automatico 2
- 20 Fai 3 interlinee verticali
- 30 Scrivi tabulando 1 "ROSSI GIOVANNI"
- 40 Fai 2 interlinee
- 50 Scrivi tabulando automatico 1 "VIA CURTATONE 55"
- 90 Scrivi tabulando 2 virgole la frase MILANO

N.B. L'allievo può sostituire questo indirizzo fittizio con il proprio

LISTATO ESERCIZIO 3

```
10 PRINT,,"BASIC SCHOOL 3"  
15 PRINT:PRINT:PRINT  
20 PRINT,"EGR. SIG."  
30 PRINT  
40 PRINT,"ROSSI GIOVANNI"  
50 PRINT:PRINT  
60 PRINT,"VIA CURTATONE 55"  
70 PRINT  
80 PRINT,"20052 MONZA"  
90 PRINT  
100 PRINT,,"MILANO"
```

LETTURA ESERCIZIO 3

- 10 Scrivi la frase "BASIC SCHOOL 3" tabulando automatico 2
- 15 Spazia in verticale di 3 interlinee
- 20 Scrivi la frase EGR. SIG. tabulando automatico 1
- 30 Spazia una interlinea
- 40 Scrivi la frase ROSSI GIOVANNI tabulando automatico 1
- 50 Spazia 2 interlinee in verticale

LISTATO ESERCIZIO 4

```
10 PRINT,, "BASIC SCHOOL 4"  
20 PRINT:PRINT  
30 PRINT "CALCOLO AREA RETTANGOLO"  
40 PRINT:PRINT  
50 PRINT "LUNGHEZZA="  
60 REM INGRESSO DATO VARIABILE  
70 INPUT LU  
80 PRINT  
90 PRINT "LARGHEZZA="  
100 INPUT LA  
110 PRINT  
120 A=LU*LA  
130 PRINT "AREA      =" A  
140 REM LEGGI:SCRIVI AREA E DI SEGUITO  
    IL VALORE CALCOLATO IN A  
150 END
```

LETTURA ESERCIZIO 4

- 50 Scrivi la frase "LUNGHEZZA="
- 60 Ricorda che in questo punto occorre introdurre (INPUT) il dato variabile o VARIABILE 70 Tieni uno spazio in memoria per la variabile numerica LU il cui valore ti verrà fornito durante l'esecuzione del programma
- 80 Spazia in verticale 1 interlinea
- 90 Scrivi la frase "LARGHEZZA="
- 100 Tieni uno spazio in memoria per la variabile numerica LA
- 110 Spazia 1 in verticale
- 120 Fai il prodotto $LU * LA$ e chiamalo con la nuova variabile A (area)
- 130 Scrivi la frase "AREA=" e di seguito il valore A (area) calcolato in precedenza
- 150 Chiudi il programma

LISTATO ESERCIZIO 5

```
10 PRINT,,"BASIC SCHOOL 5"  
20 PRINT:PRINT  
30 PRINT,"CALCOLO AREA RETTANGOLO"  
40 PRINT:PRINT:PRINT  
50 REM SCRIVI:LUNGHEZZA IN MT.= E INTRODUCI  
   IL DATO VARIABILE INDICATO CON LU  
60 INPUT "LUNGHEZZA MT.           ";LU  
70 PRINT  
80 INPUT "LARGHEZZA MT.           ";LA  
90 PRINT:PRINT  
100 A=LU*LA  
110 PRINT "AREA DEL RETTANGOLO MQ.  " A  
120 PRINT:PRINT:PRINT  
130 END
```

LETTURA ESERCIZIO 5

- 10 Scrivi la frase BASIC SCHOOL tabulando 2 automatico
- 20 Interlinea di 2 spazi in verticale.
- 30 Scrivi la frase "CALCOLO AREA RETTANGOLO" tab. autom. 1
- 40 Spazia 3 interlinee in verticale.
- 50 Ricorda di scrivere "LUNGHEZZA MT" e di introdurre il dato variabile (o semplicemente la VARIABILE) relativa
- 60 Scrivi sullo schermo "LUNGHEZZA MT" e attribuisce (;) alla lunghezza la sigla LU
- 70 Uno spazio in verticale
- 80 Scrivi sullo schermo "LARGHEZZA IN MT" e attribuisce (;) alla larghezza la sigla LA
- 90 Spazia 2 righe in verticale
- 100 Esegui il prodotto A (area) =LU * LA. A è la sigla di una nuova variabile numerica il cui valore è interno al calcolatore e varia al variare di LU e LA
- 110 Scrivi la frase "AREA DEL RETTANGOLO MQ= " e, subito di seguito, il valore di A che hai calcolato con la riga 100
- 120 Spazia 3 righe prima di chiudere
- 130 Chiudi il programma

Al termine dell'esercizio controllare che l'inizio dei numeri risulti allineato

LISTATO ESERCIZIO 6

```
10 PRINT,, "BASIC SCHOOL 6"  
20 PRINT:PRINT  
30 PRINT TAB(7)"CONDIZIONALE E SALTO"  
40 PRINT:PRINT:PRINT:PRINT  
50 REM AZZERAMENTO VARIABILI  
60 LU=0:LA=0  
70 INPUT "LUNGHEZZA          . =";LU  
80 PRINT  
90 INPUT "LARGHEZZA          =";LA  
100 PRINT  
110 A=LU*LA  
120 PRINT"AREA RETTANGOLO      =" A  
130 PRINT:PRINT:PRINT:PRINT  
140 INPUT "VUOI RIPETERE      S/N";R$  
150 REM CONDIZIONE PER SALTARE  
160 IF R$="S" THEN PRINT"☐": GOTO 10  
170 IF R$="N" THEN END  
180 END  
190 REM "☐" E' IL SEGNO GRAFICO DI "SHIFT/CLR/HOME"
```

LETTURA ESERCIZIO 6

- 40 Spazia 4 battute verticali
- 60 Azzeri i valori delle variabili LU e LA
- 70 Scrivi la frase "LUNGHEZZA =" e fai il punto di domanda per una variabile numerica alla quale attribuirai (;) la sigla LU
- 80 Spazia 1 battuta in verticale
- 90 Scrivi la frase "LARGHEZZA =" e fai il punto di domanda per una variabile numerica alla quale attribuirai (;) la sigla LA
- 100 Spazia 1 battuta in verticale
- 110 Calcola $A = LU * LA$
- 120 Scrivi la frase "AREA RETTANGOLO =" e, di seguito, il valore calcolato in A
- 140 Fai il punto interrogativo per una variabile stringa (quindi non numerica pura). Attribuiscele (;) la sigla R\$ (si legge R dollaro oppure R stringa. È stata scelta la R perchè è l'iniziale di Ripetere)
- 160 Se R\$ è = alla lettera "S" (mettere fra gli apici), ripulisci lo schermo e inizia dalla linea di programma n. 10 (GOTO = VAI A)
- 170 Se R\$ è = alla lettera "N", chiudi il programma
- 180 Chiudi il programma
- 190 Ricorda che PRINT "☐" dà istruzione al calcolatore perchè svolga dal suo interno la stessa funzione che si ottiene premendo contemporaneamente i tasti SHIFT e CLR/HOME

LISTATO ESERCIZIO 7

```
10 REM RIPULISCI LO SCHERMO E PARTI DALL'ALTO
20 PRINT"☐"
30 PRINT,,"BASIC SCHOOL 7"
40 PRINT:PRINT:PRINT
50 PRINT,"SCELTA CONDIZIONATA"
60 PRINT:PRINT:PRINT
65 LU=0:LA=0
70 INPUT "LUNGHEZZA           =" ;LU
80 PRINT
90 INPUT "LARGHEZZA           =" ;LA
100 PRINT
110 A=LU*LA
120 PRINT "AREA DEL RETTANGOLO =" A
130 PRINT:PRINT
140 REM CONDIZIONI PER SCRIVERE
150 IF A<300 THEN PRINT"VALORE INFERIORE A 300"
160 IF A=300 THEN PRINT"300 E' IL VALORE OK"
170 IF A>300 THEN PRINT"VALORE SUPERIORE A 300"
180 PRINT:PRINT:PRINT
190 INPUT"#####RIPETERE     S/N";R$
220 REM CONDIZIONE PER RIPETERE
230 IF R$="S" THEN 10
240 IF R$<>"S" THEN END
250 REM "☐" E' IL SEGNO GRAFICO DEL CURSOR DESTRA
```

LETTURA ESERCIZIO 7

- 10 Ricorda di pulire lo schermo con PRINT "☐"
- 20 Pulisci lo schermo e parti dall'alto.
- 150 Se il valore di A (area) che hai calcolato è inferiore a 300 scrivi sullo schermo la frase "VALORE INFERIORE A 300"
- 160 Se A = 300 scrivi la frase "300 È IL VALORE OK"
- 170 Se A è superiore a 300 scrivi la frase "VALORE SUPERIORE A 300"
- 180 Spazia 3 interlinee in verticale
- 190 Scrivi la frase "RIPETERE S/N" tabulando 11 battute a destra (☐ è il segno grafico di CRSR destra) e fai il punto di domanda cui ti verrà risposto con un dato stringa (quindi non numerico puro). Attribuisce alla variabile la sigla R\$ (R dollaro)
- 230 Se R\$ = alla lettera S (cioè se alla domanda rispondo con la lettera S), vai alla riga 10 (cioè ripeti il programma)
N.B. THEN 10=THEN GOTO 10
- 240 Se R\$ è diverso dalla lettera S (cioè se alla domanda rispondo con un'altra lettera oppure con un numero o con altro segno), allora chiudi il programma

LISTATO ESERCIZIO 8

```
10 REM RIPULISCI LO SCHERMO E PARTI DALL'ALTO
20 PRINT"␣"
30 PRINT,,"BASIC SCHOOL 8"
40 PRINT:PRINT:PRINT
45 LU=0:LA=0 :REM AZZERAMENTO VARIABILI
50 PRINT,"RIPETIZIONE COL GET"
60 PRINT:PRINT:PRINT
70 INPUT "▀▀LUNGHEZZA           =" ;LU
80 PRINT
90 INPUT "▀▀LARGHEZZA           =" ;LA
100 PRINT
110 A=LU*LA
120 PRINT "▀▀AREA DEL RETTANGOLO =" A
130 PRINT:PRINT
140 REM CONDIZIONI PER SCRIVERE
150 IF A<300 THEN PRINT TAB(10) "VALORE
    INFERIORE A 300"
160 IF A=300 THEN PRINT TAB(10)"300 E' IL
    VALORE OK"
170 IF A>300 THEN PRINT TAB(10)"VALORE
    SUPERIORE A 300"
180 PRINT:PRINT:PRINT:PRINT
190 PRINT TAB(10) "RIPETERE           S/N"
220 REM CONDIZIONE PER RIPETERE
250 REM "▀" E' IL SEGNO GRAFICO DEL CURSOR
    DESTRA
260 GET R$
270 IF R$ = "S" THEN 10
280 IF R$ = "N" THEN END
290 GOTO 260
```

LETTURA ESERCIZIO 8

- 45 Azzera la variabili LU e LA
- 60 Spazia 3 interlinee in verticale
- 70 Scrivi la frase "LUNGHEZZA =", tabulando 2 battute con il CURSORE A DESTRA (dopo aver fatto l'apice, battere il tasto CASA→).
Attribuisci alla variabile la sigla LU

- 90** Idem come in 70, ma la frase da scrivere è “LARGHEZZA =” e la variabile è LA
- 120** Scrivi tabulando 2 battute con il CURSORE A DESTRA la frase “AREA RETTANGOLO =” e, accanto, il valore di A
- 130** 2 spazi in verticale
- 150** Se il valore calcolato in A (area) è < 300 scrivi tabulando 10 la frase “VALORE INFERIORE A 300”
- 160** Se il valore calcolato in A è $= 300$ scrivi tabulato 10 la frase “300 E’ IL VALORE OK”
- 170** Se il valore di A > 300 allora scrivi tabulando 10 la frase “VALORE SUPERIORE A 300”
- 180** Spazia 4 interlinee
- 190** Scrivi tabulando 10 battute la frase “RIPETERE S/N”
- 260** Attendi un comando di tastiera per la variabile R\$ (R dollaro = R stringa)
- 270** Se sulla tastiera verrà battuta la lettera “S” vai alla riga n. 10 (cioè ripeti il programma)
- 280** Se sulla tastiera verrà battuta la lettera “N”, chiudi il programma
- 290** Ritorna alla riga d’inizio del GET (istruzione obbligatoria)

LISTATO ESERCIZIO 9

```
10 PRINT"□"  
20 PRINT,,"BASIC SCHOOL 9"  
30 PRINT:PRINT:PRINT  
40 PRINT,"OPERAZIONE CICLICA"  
45 PRINT,"-----"  
50 PRINT:PRINT:PRINT  
60 PRINT TAB(6)"STAMPARE I NUMERI DA 1 A 20"  
65 PRINT:PRINT:PRINT  
70 FOR K=1 TO 20  
80 PRINT K;  
90 NEXT K  
100 REM IL ; DOPO UNA RIGA DI PRINT IMPEDISCE  
    AL CURSORE DI ANDARE A CAPO  
110 PRINT:PRINT:PRINT  
120 REM SOMMATORIA DEI NUMERI  
130 FOR K=1 TO 20  
140 S=S+K  
150 NEXT K  
160 PRINT TAB(6)"LA SOMMA DEI NUMERI E'";S  
170 PRINT
```

LETTURA ESERCIZIO 9

- 10 Pulisci lo schermo e inizia dall'alto.
- 40 Scrivi tabulando automatico 1 (10 battute) la frase "OPERAZIONE CICLICA"
- 45 Stampa tabulando automatico 1 (10 battute) la linea grafica "□" 10 volte) ottenuta battendo simultaneamente i tasti **Q** e **Y**
- 50 3 interlinee di spazio verticale
- 70 Opera ciclicamente attribuendo alla variabile K in progressione i valori che vanno da 1 a 20, cioè, prima dà a K il valore 1; dopo il NEXT, K=2, dopo il successivo NEXT, K=3 ecc
- 80 Scrivi uno accanto all'altro i valori assunti da K durante i diversi cicli
- 90 Punto di chiusura per la ripetizione di ogni ciclo
- 120 Ricorda di sommare tra di loro i diversi valori assunti progressivamente dalla variabile K
- 130 Per K che va da 1 a 20
- 140 La somma è ottenuta con la formula $S=S+K$
- 150 Chiusura di ogni ciclo
- 160 Scrivi sullo schermo tabulando 6, la frase "LA SOMMA DEI NUMERI È" e, di seguito, scrivi il valore calcolato per S nella riga 140
- 170 Spazia in verticale due battute

LISTATO ESERCIZIO 10

```
10 PRINT"3"
20 PRINT,,"BASIC SCHOOL 10"
30 REM SUBROUTINE PER SPAZIARE IN VERTICALE
40 GOSUB 230
50 PRINT TAB(6)"OPERAZIONE CICLICA A PASSO"
55 PRINT TAB(6)"-----"
60 GOSUB 230
70 PRINT TAB(6)"MOLTIPLICARE 40 * I NUMERI"
80 PRINT TAB(6)"DA 1 A 20 SALTANDO PER 2"
90 GOSUB 230
100 FOR K=1 TO 20 STEP 2
110 P=40*K
120 S=S+P
130 PRINT P;
140 NEXT K
150 GOSUB 230
160 PRINT TAB(6)"LA SOMMA DEI PRODOTTI E' " S
170 PRINT
180 END
190 REM SUBROUTINE PER SPAZIARE 3. FINISCE
    CON RETURN
230 FOR L=1 TO 3
240 PRINT
250 NEXT L
260 RETURN
```

LETTURA ESERCIZIO 10

- 10 Pulisci lo schermo e inizia dall'alto
- 30 Ricorda di fare la subroutine (sottoprogramma) della riga 230
- 40 Salta alla subroutine della riga 230
- 50 Scrivi tabulando 6 battute la frase "OPERAZIONE CICLICA A PASSO"
- 55 Stampa tabulando 6 battute il segno grafico sotto tutta la frase precedente (il segno "" è ottenuto premendo **⌘** e la lettera Y)
- 60 Salta alla subroutine della riga 230
- 70 Scrivi tabulando 6 la frase "MOLTIPLICARE 40* I NUMERI"
- 80 Scrivi tabulando 6 la frase "DA 1 A 20 SALTANDO PER 2"
(N.B. La frase esposta sopra è stata divisa in due per problemi di composizione grafica)

- 90** Salta alla subroutine della riga 230
- 100** Per K che varia da 1 a 20, a passo di 2
- 110** Calcola ad ogni ciclo il prodotto $P=40 * K$
- 120** Fai la sommatoria dei prodotti con la formula $S=S+P$
- 130** Scrivi ad ogni ciclo il valore dei prodotti P, ponendoli uno accanto all'altro (sviluppo orizzontale)
- 140** Chiusura di ogni ciclo
- 150** Vai alla riga 230 per la subroutine di spaziamento verticale
- 160** Scrivi tabulando 6 la frase "LA SOMMA DEI PRODOTTI È" e accanto scrivi il valore finale di S (perchè la riga 160 è esterna al ciclo FOR NEXT)
- 170** Spazia una battuta in verticale prima del READY.
- 180** Chiudi il programma.
- 190** Ricorda di fare la subroutine (sottoprogramma) per spaziare 3 battute in verticale
- 230** Per L=1 a 3
- 240** Fai uno spazio in verticale
- 250** Chiudi il ciclo
- 260** Ritorna alla riga di programma che segue l'istruzione GOSUB

LISTATO ESERCIZIO 11

```
5 PRINT "3"
10 PRINT,, "BASIC SCHOOL 11"
20 SP=2:GOSUB 150
30 PRINT TAB(6) "OPERAZIONE CICLICA INVERSA"
40 PRINT TAB(6) "-----"
50 SP=1:GOSUB 150
60 PRINT TAB(6) "MOLTIPLICARE 40 * I NUMERI"
70 PRINT TAB(6) "DA 20 A 1 SALTANDO PER 2"
80 SP=3:GOSUB 150
90 FOR K= 20 TO 1 STEP -2
100 P=40*K
110 S=S+P
120 PRINT TAB(4)"PER K ="K TAB(17)"P ="P
    TAB(27)"S ="S
130 NEXT K
140 END
150 FOR L=1 TO SP:PRINT:NEXT L:RETURN
```

LETTURA ESERCIZIO 11

- 5 Pulisci lo schermo e inizia dall'alto
- 10 Scrivi la frase "BASIC SCHOOL 11" tabulando 20 battute
- 20 Per fare SP=2 spazi, salta alla subroutine della riga 150
- 30 Scrivi, tabulando 6, la frase "OPERAZIONE CICLICA INVERSA"
- 40 Stampa, tabulando 6, il segno ripetendolo per tutta la lunghezza della frase di cui alla riga 30. Il segno lo si ottiene premendo **☉** e il tasto Y
- 50 SP=1, salta alla subroutine della riga 230
- 90 Per K che va da 20 a 1, a passo di -2
- 100 Calcola ad ogni ciclo il prodotto $P = 40 * K$
- 110 Calcola ad ogni ciclo la somma dei prodotti $S = S + P$
- 120 Scrivi tab. 4 la frase "PER K=" e di seguito il valore di K, poi tab. 17 la frase "P=" e di seguito il valore di P, poi ancora la frase "S=" seguita dal valore di S
In pratica, compariranno le seguenti frasi:
PER K=20 P=800 S= 800
PER k=18 P=720 S=1520
ecc. ecc.
- 130 Chiudi il ciclo del FOR K col NEXT K

- 140** Chiudi il programma
- 150** Per L che va da 1 a SP, fai uno spazio in verticale, chiudi col NEXT L quindi ritorna alla riga di programma che segue il GOSUB
- 200** Chiudi il ciclo del FOR con il NEXT
- 210** Ritorna alla riga che segue il GOSUB

LISTATO ESERCIZIO 12

```
10 PRINT"0"  
20 PRINT,,"BASIC SCHOOL 12"  
30 PRINT:PRINT  
40 PRINT,"DATARIO CON IF...THEN"  
50 PRINT,"-----"  
60 REM SOTTOLINEARE CON COMMODORE E Y  
70 PRINT:PRINT  
80 FOR K=1 TO 7  
85 PRINT TAB(15)  
90 IF K=1 THEN PRINT "LUNEDI"  
100 IF K=2 THEN PRINT "MARTEDI"  
110 IF K=3 THEN PRINT "MERCOLEDI"  
120 IF K=4 THEN PRINT "GIOVEDI"  
130 IF K=5 THEN PRINT "VENERDI"  
140 IF K=6 THEN PRINT "SABATO"  
150 IF K=7 THEN PRINT "DOMENICA"  
160 PRINT  
170 NEXT K
```

LETTURA ESERCIZIO 12

- 10 Pulisci lo schermo e inizia dall'alto
- 20 Scrivi "BASIC SCHOOL 12", tabulando automatico 2
- 30 Spazia 2 interlinee in verticale
- 40 Scrivi, tabulando autom. 1, la frase DATARIO CON IF...THEN
- 50 Stampa tabulando autom. 1, la sottolineatura sotto la frase precedente
- 60 Ricorda per sottolineare premere Commodore e Y
- 70 2 spazi di interlinea
- 80 Per 7 volte (K da 1 a 7) fai un ciclo FOR...NEXT
- 85 Ad ogni ciclo tabula di 15 battute
- 90 Se K = 1 scrivi la frase LUNEDI
- 100 Se K = 2 scrivi la frase MARTEDI
- 110 Se K = 3 scrivi la frase MERCOLEDI
- 120 Se K = 4 scrivi la frase GIOVEDI
- 130 Se K = 5 scrivi la frase VENERDI
- 140 Se K = 6 scrivi la frase SABATO
- 150 Se K = 7 scrivi la frase DOMENICA
- 160 Fai uno spazio verticale prima di chiudere ogni ciclo
- 170 Chiudi il ciclo del FOR K col NEXT K

LISTATO ESERCIZIO 13

```
10 PRINT"3"  
20 PRINT,,"BASIC SCHOOL 13"  
30 PRINT:PRINT  
40 PRINT,"DATARIO USANDO ON"  
50 PRINT,"-----"  
60 REM SOTTOLINEARE CON COMMODORE E Y  
70 PRINT:PRINT  
80 INPUT "N.GIORNO";N  
83 PRINT:PRINT  
85 ON N GOTO 90,100,110,120,130,140,150  
87 IF N<1 OR N>7 THEN END  
90 PRINT "LUNEDI";:GOTO 160  
100 PRINT "MARTEDI";:GOTO 160  
110 PRINT "MERCOLEDI";:GOTO 160  
120 PRINT "GIOVEDI";:GOTO 160  
130 PRINT "VENERDI";:GOTO 160  
140 PRINT "SABATO";:GOTO 160  
150 PRINT "DOMENICA" ;  
160 PRINT " E' IL"N"GIORNO DELLA SETTIMANA"  
170 GOTO 70
```

LETTURA ESERCIZIO 13

- 40 Scrivi (tabul. autom. 1) la frase DATARIO USANDO ON
- 80 Scrivi la frase "N. GIORNO" e crea un Input per la variabile numerica N
- 85 In corrispondenza dei valori progressivi (1...7) che attribuisco a N, salta alle righe di programma 90,100,110,120,130,140,150
- 90 Scrivi la frase LUNEDI e, accanto, scrivi quanto stabilito dalla riga 160
- 100 Idem per MARTEDI
- 110 Idem per MERCOLEDI
- 120 Idem per GIOVEDI
- 130 Idem per VENERDI
- 140 Idem per SABATO
- 150 Idem per DOMENICA
- 160 Scrivi la frase E' IL, poi di seguito il valore attribuito a N, quindi sempre accanto, la frase GIORNO DELLA SETTIMANA
- 170 Ritorna alla riga 70, cioè ripeti la richiesta del numero N dopo aver spaziato 2 righe in verticale

LISTATO ESERCIZIO 14

```
10 PRINT"3"  
20 PRINT,"BASIC SCHOOL 14"  
30 PRINT:PRINT  
40 PRINT TAB(8)"DATARIO CON READ APERTO"  
50 PRINT TAB(8)"_____  
60 PRINT  
70 REM LEGGI LA VARIABILE STRINGA IN SEQUENZA  
80 READ G$  
90 REM ELENCO DATI DA LEGGERE  
100 DATA LUNEDI,MARTEDI,MERCOLEDI,GIOVEDI,  
    VENERDI,SABATO,DOMENICA  
110 PRINT TAB(9)"SETTEMBRE  " G$  
120 PRINT  
130 GOTO 80  
140 REM RITORNA AL READ AD OGNI PASSAGGIO
```

LETTURA ESERCIZIO 14

- 10 Pulisci lo schermo e inizia dall'alto
- 80 Leggi la variabile G\$ (G STRINGA) (si tratta di dati non numerici)
- 90 Elenco dati stringa da leggere ad ogni passaggio: LUNEDI, MARTE-
DI, ecc. ecc.
- 100 Scrivi tabulando 9 battute la frase "SETTEMBRE", seguita dalla stringa G\$ letta in questo passaggio
- 120 Fai uno spazio in verticale
- 130 Ritorna alla riga 80 per leggere il dato successivo

LISTATO ESERCIZIO 15

```
10 PRINT "3"
20 PRINT , "BASIC SCHOOL 15"
30 PRINT : PRINT
40 PRINT TAB(7) "DATARIO CON READ CHIUSO"
50 PRINT TAB(7) "-----"
60 PRINT : PRINT
70 FOR K= 18 TO 24
80 READ G$
90 REM ELENCO STRINGHE DA LEGGERE
100 DATA LUNEDI, MARTEDI, MERCOLEDI, GIOVEDI,
    VENERDI, SABATO, DOMENICA
110 PRINT TAB(6) K "SETTEMBRE  -" SPC(2) G$
120 PRINT
130 NEXT K
140 REM RITORNA AL READ AD OGNI PASSAGGIO
```

LETTURA ESERCIZIO 15

- 10 Pulisci lo schermo e inizia dall'alto
- 70 Per 7 volte (K che va da 18 a 24)
- 80 Leggi la variabile stringa G\$
- 100 Elenco delle stringhe da leggere: LUNEDI, MARTEDI, ecc. ecc.
- 110 Scrivi, tabulando 6, il valore di K (quindi 18,19,20 ecc.) seguito dalla frase "SETTEMBRE —" e, spaziando 2 battute (SPC(2)) dalla stringa G\$ letta in questo passaggio
- 120 Fai uno spazio in verticale
- 130 Chiudi il ciclo e ritorna alla riga 70

LISTATO ESERCIZIO 16

```
10 PRINT"J"  
20 PRINT,,"BASIC SCHOOL 16"  
30 PRINT:PRINT  
35 PRINT TAB(6)"PROVA DI READ CON 10 NUMERI"  
40 PRINT TAB(6)"-----"  
50 PRINT  
60 FOR K=1 TO 10  
70 READ P  
80 DATA 45,89,5,67.8,1,456,23,78,89,4  
90 PRINT TAB(10) P  
100 S=S+P  
110 NEXT K  
120 PRINT:PRINT  
130 PRINT TAB(2)"LA SOMMA DI TUTTI I NUMERI  
    E'=" S  
140 PRINT  
150 PRINT TAB(2)"IL VALORE MEDIO E'=" S/(K-1)
```

LETTURA ESERCIZIO 16

- 10 Pulisci lo schermo e comincia dall'alto
- 60 Per 10 cicli (K da 1 a 10)
- 70 Leggi la variabile numerica P
- 80 Dati da leggere in sequenza: 45,89,5,67.8,1,456,23,78,89,4
- 90 Scrivi, tabulando 10, il valore della variabile P
- 100 Fai la sommatoria dei dati con la formula $S=S+P$
- 110 Chiudi il ciclo K
- 120 Spazia 2 in verticale
- 130 Scrivi, tabulando 2, la frase "LA SOMMA DI TUTTI I NUMERI È=" e di seguito scrivi il valore di S (valore finale della sommatoria dei dati, finale perchè è preso al termine di tutti i cicli quando si sono aggiunti tutti i valori che costituiscono l'insieme dei dati)
- 140 Spazia una battuta in verticale
- 150 Scrivi, tab. 2, la frase "IL VALORE MEDIO È=" e di seguito il valore ricavato dalla divisione $S/(K-1)$, cioè il valore medio dei dati letti

LISTATO ESERCIZIO 17

```
5 PRINT "J"
10 PRINT,,"BASIC SCHOOL 17"
20 PRINT:PRINT
30 PRINT TAB(7) "OPERAZIONE CICLICA DOPPIA"
40 PRINT TAB(7) "-----"
50 PRINT
60 PRINT TAB(6) "CIASCUN NUMERO DA 15 A 20"
70 PRINT TAB(6) "VA MOLTIPLICATO DA 1 A 5"
80 PRINT:PRINT
90 FOR K= 15 TO 20
95 PRINT TAB(5)
100 FOR J= 1 TO 5
110 P=K*J
120 S=S+P
130 PRINT P SPC(2);
140 NEXT J
145 PRINT
150 NEXT K
160 PRINT:PRINT
170 PRINT TAB(2) "LA SOMMA DI TUTTI I
    PRODOTTI E'"S
180 PRINT
190 PRINT TAB(2)"LA MEDIA DI TUTTI I PRODOTTI
    E'"S/30
```

LETTURA ESERCIZIO 17

- 10 Pulisci lo schermo e parti dall'alto
- 70 Scrivi, tabulando 6, la frase "CIASCUN NUMERO DA 15 A 20"
- 80 Scrivi, tabulando 6, la frase "VA MOLTIPLICATO DA 1 A 5"
N.B. Si tratta di una frase unica spezzata in due per motivi di armonia compositiva
- 90 FOR K = 15 TO 20...NEXT K rappresenta il ciclo esterno
- 95 Tabula di 5 battute prima di entrare nel cerchio interno
- 100 FOR J = 1 TO 5...NEXT J rappresenta il ciclo interno

- 110** Calcola i prodotti $P = K * J$
120 Calcola la sommatoria $S = S+P$
130 Scrivi, tabulando 4 battute, il valore dei prodotti P, poi spazia 2 battute e ferma il cursore (;)
140 Chiudi il ciclo interno di J
145 Manda a capo il cursore
150 Chiudi il ciclo esterno di K
160 Fai 2 spazi in verticale
170 Scrivi, tab. 2, la frase “LA SOMMA DI TUTTI I PRODOTTI È” e di seguito il valore finale di S
180 Spazia 1 interlinea
190 Scrivi, tab. 2, la frase “LA MEDIA DI TUTTI I PRODOTTI È” e di seguito stampa il valore di M calcolato nella riga 200 dividendo S per 30
N.B. Il numero 30 è stato calcolato tenendo conto che vi sono 6 valori di K e cioè 15-16-17-18-19 e 20 e 5 valori di J e cioè 1-2-3-4-5. In totale quindi sono $6*5=30$

LISTATO ESERCIZIO 18

```
10 PRINT"3"
20 PRINT,,"BASIC SCHOOL 18"
30 PRINT:PRINT
40 PRINT TAB(8)"MATRICE CON DATI INTERNI"
50 PRINT TAB(8)"-----"
60 PRINT:PRINT
65 PRINT TAB(2)"I DATI FORNITI SONO:"
67 PRINT
70 FOR I=1 TO 8
80 READ D(I)
90 DATA 23,4,789,12.3,1,67,58,34.8
105 PRINT TAB(11)D(I)
110 NEXT I
120 PRINT
125 S=D(1)+D(8) : ME=D(3)/2
130 PRINT TAB(2)"LA SOMMA DEL PRIMO CON
    L'ULTIMO:" S
150 PRINT TAB(2)"LA META' DEL DATO N.3 :" ME
```

LETTURA ESERCIZIO 18

- 10 Pulisci lo schermo e ricomincia dall'alto
- 20 Scrivi tabulando automatico 2 la frase "BASIC SCHOOL 18"
- 30 Spazia 2 interlinee.
- 40 Scrivi, tabulando 8, la frase "MATRICE CON DATI INTERNI"
- 50 Stampa, tabulando 8, la linea fornita dai tasti $\text{Ctrl} + Y$
- 60 Spazia 2 battute in verticale
- 70 Per I (indice) che varia da 1 a 8
- 80 Leggi la variabile indicizzata D(I)
- 90 I dati sono: 23,4,789,12.3,1,67,58,34,8
- 105 Scrivi, tabulando 11, il valore della variabile D(I)
- 110 Chiudi il ciclo di I (variabile indice)
- 120 Uno spazio verticale
- 125 Fai le operazioni $S=D(1)+D(8)$ e $ME=D(3)/2$
- 130 Scrivi, tabulando 2, la frase "LA SOMMA DEL PRIMO CON L'ULTIMO:" e accanto stampa il valore di S calcolato alla riga 125
- 150 Scrivi tab. 2 la frase "LA META' DEL DATO N. 3" e di seguito il valore di ME calcolato alla riga 125

LISTATO ESERCIZIO 19

```
10 PRINT"J"
20 PRINT,"BASIC SCHOOL 19"
30 PRINT:PRINT
40 PRINT TAB(8)"MATRICE CON DATI ESTERNI"
50 PRINT TAB(8)"-----"
60 PRINT:PRINT
65 PRINT TAB(4)"INTRODUCI 7 NUMERI : "
70 PRINT"J";REM RISALI DI 1 BATTUTA
75 REM INTRODUZIONE DATI
80 FOR I=1 TO 7
90 INPUT "!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!";D(I)
100 NEXT I
110 PRINT:PRINT
115 REM LETTURA DATI PER IL CALCOLO DELLA
    SOMMATORIA S
120 FOR I=1 TO 7
130 S=S+D(I)
140 NEXT I
150 PRINT TAB(4)"LA SOMMA DEI NUMERI E' : " S
160 PRINT
170 M=S/7
180 PRINT TAB(4)"IL VALORE MEDIO E'      : " M
190 PRINT
200 TS=D(3)+D(6)
210 PRINT TAB(4)"IL TERZO + IL SESTO    : " TS
```

LETTURA ESERCIZIO 19

- 65 Scrivi, tabulando 4, la frase "INTRODUCI 7 NUMERI"
- 70 Risali col cursore di una battuta, cioè sulla riga della frase di cui al 65. Per risalire usare l'istruzione PRINT "█"; . Dopo il primo apice, premere SHIFT +CASA T, quindi fare il secondo apice e il ;
- 80 Con l'indice I che va da 1 a 7 (quindi per 7 volte).
- 90 Fai un punto di domanda (INPUT) dopo aver portato il cursore a destra di 22 battute. Attribuisce alla variabile con indice la sigla D(I) valida per dati numerici
- 100 Chiudi il ciclo di lettura (FOR...NEXT I).

- 120** Con l'indice I che va da 1 a 7 (quindi per 7 volte).
- 130** Leggi i dati e calcola la sommatoria $S = S+D(I)$
- 140** Chiudi il ciclo di lettura
- 150** Scrivi, tabulando 4, la frase "LA SOMMA DEI NUMERI È:" e di seguito scrivi il valore calcolato in S (riga 130).
- 160** Fai uno spazio verticale.
- 170** Calcola il valore medio dei dati con l'espressione $M = S/7$ oppure $M = S/(I-1)$, dato che al termine dell'ultimo ciclo I assume il valore di 8
- 180** Scrivi, tabulando 4, la frase "IL VALORE MEDIO È : " e, di seguito, stampa il valore di M calcolato con la riga 170
- 190** Fai uno spazio verticale
- 200** Calcola la somma $TS=D(3)+D(6)$
- 210** Scrivi, tabulando 4, la frase "IL TERZO + IL SESTO : " e, di seguito, il valore calcolato per TS alla riga 200

LISTATO ESERCIZIO 20

```
5 PRINT "3"
10 PRINT ,,"BASIC SCHOOL 20"
20 PRINT
30 PRINT TAB(10)"MATRICE CON MASCHERA"
40 PRINT TAB(10)"-----"
50 PRINT
60 PRINT TAB(5)"QUANTI DATI? (MAX. 10):"
70 PRINT
80 PRINT TAB(5)"I DATI:"
90 FOR K=1 TO 10:PRINT:NEXT K
100 PRINT TAB(5)"IL PRIMO + L'ULTIMO : "
110 FOR K=1 TO 40:PRINT"-";:NEXT K:PRINT
120 PRINT TAB(8)"RIPETERE S/N"
130 FOR K=1 TO 17:PRINT CHR$(145);:NEXTK
140 PRINT TAB(27)
150 N=0: INPUT N
160 IF N=0 THEN PU=0 :GOTO 220
170 PRINT
180 FOR I=1 TO N
190 PRINT TAB(16)
200 D(I)=0 : INPUT D(I) : NEXT I
210 IF N=1 THEN PU=D(1): GOTO 220
215 PU=D(1)+D(N)
220 PRINT "8"
230 FOR K=1 TO 18: PRINT: NEXTK
240 PRINT TAB(28)PU
250 PRINT
260 PRINT TAB(27)
270 INPUT R$
280 IF R$<>"S" THEN END
290 IF R$="S" THEN 5
```

LETTURA ESERCIZIO 20

5 - 120 CREAZIONE DEL PRINTATO O MASCHERA

130 Per 17 volte (K da 1 a 17), sposta il cursore in alto di 1 battuta, una volta di seguito all'altra (;) poi chiudi il ciclo. Riassumendo, sposta il cursore in alto (PRINT CHR\$(145)) per 17 battute

- 140** Sposta il cursore a destra di 27 battute
- 150** Crea un INPUT per la variabile numerica N dopo averla azzerata
- 160** Se rispondo con lo zero, anche $PU=0$, vai alla riga 220
- 170** Fai uno spazio verticale
- 180** Per I= da 1 al valore finale attribuito ad N alla riga 150
- 190** Sposta il cursore a destra di 16 battute.
- 200** Crea un INPUT (ingresso dall'esterno) per la variabile numerica indicizzata D(I), dopo averla preventivamente azzerata e chiudi il ciclo
- 210** Se $N=1$, PU sarà uguale al primo dato D(1), poi vai a 220
- 215** Fai la somma $PU=D(1)+D(N)$, cioè somma il 1° dato D(1) con l'ultimo dato D(N)
- 220** Riporta il cursore in alto a sinistra nella posizione HOME con l'espressione PRINT "␣". Dopo il primo apice, premere HOME
- 230** Per 18 volte riporta il cursore in basso di 1 battuta (18 spazi in verticale)
- 240** Sposta il cursore a destra di 28 battute e scrivi il valore di PU calcolato in 215
- 250** Spazia di una interlinea.
- 260** Tabula il cursore di 27 battute.
- 270** Crea un INPUT. Attribuisce alla variabile stringa la sigla R\$ (R come Ripetere, \$ come stringa)
- 280** Se la risposta è diversa dalla lettera S chiudi il programma
- 290** Se la risposta è uguale alla lettera S, ritorna alla riga 5, cioè ripeti il programma

INDICE ISTRUZIONI E COMANDI

| Descrizione | Pagina |
|-------------------------|-------------|
| CLOSE | 99 |
| CMD | 100 |
| DIM | 71 |
| END | 42 |
| FOR...NEXT | 37 |
| FOR...NEXT...STEP | 38 |
| GET | 36 |
| GOSUB | 41 - 45 |
| GOTO | 31 |
| IF...THEN | 31 |
| INPUT | 22 - 95 |
| INPUT# | 99 |
| INT | 63 |
| LIST | 7 - 95 |
| LOAD | 20 - 21 |
| NEW | 13 |
| ON...GOTO | 51 |
| OPEN | 99 |
| PRINT | 7 - 29 - 96 |
| PRINT CHR\$ | 96 |
| PRINT# | 76 |
| PRINT TAB(X) | 97 |
| READ...DATA | 53 |
| REM | 23 |
| RETURN | 7 |
| RND (1) | 105 |
| RUN | 7 - 95 |
| SAVE | 13 - 14 |
| SPC(X) | 55 |
| TAB(X) | 17 |

INDICE ARGOMENTI DI INFORMATICA

| Argomento | Pagina |
|--------------------------------------|--------------|
| ASCII | 20 |
| Algol | 49 |
| ALU | 26 |
| Assembler | 19 - 49 |
| Basic | 19 |
| Baud | 15 |
| Bit | 11 |
| Byte | 11 |
| BUS degli indirizzi | 40 |
| BUS dei dati | 40 |
| Caricamento dati e/o programmi | 20 - 21 - 59 |
| Circuito Integrato | 19 |
| Chip | 20 |
| Clock | 39 |
| Cobol | 49 |
| Commodore 64 | 10 |
| Compilatore | 50 |
| Digitale | 25 |
| Editor | 50 |
| File | 67 |
| Floppy disk | 59 |
| Formattazione | 60 |
| Fortran | 49 |
| Grafica | 25 |
| Hard disk | 12 |
| Hardware | 19 |
| IEEE | 25 |
| Indirizzi | 50 |
| Inizializzazione | 60 |
| Interfaccia | 15 |
| Lettura | 99 |
| Logico | 25 |
| Loop | 37 |
| Main Frame | 40 |
| Maschera | 69 |
| Matrici lineari | 65 |
| Memoria di massa | 9 |
| Modem | 16 |
| Monitor video | 10 |
| Multiprogrammazione | 40 |
| Office Automation | 16 |

INDICE ARGOMENTI DI INFORMATICA

| Argomento | Pagina |
|----------------------------------|--------------|
| Pascal | 49 |
| Personal Computer | 58 |
| Plotter | 25 |
| RAM | 10 - 50 |
| ROM | 11 |
| Salvataggio dati | 67 - 80 - 99 |
| Scrittura | 67 |
| Semigrafica | 25 |
| Sistema operativo | 10 |
| Software | 12 - 19 |
| Stampante | 9 - 25 |
| Stringa | 31 |
| Subroutine | 41 - 75 |
| Tastiera | 6 |
| Triumph Adler | 58 |
| Trasmissione parallela | 15 |
| Trasmissione seriale | 15 |
| Variabile numerica | 22 |
| Variabile numerica intera | 42 |
| Variabile numerica stringa | 23 |
| Word Processing | 16 - 58 - 60 |

ISTRUZIONI PER IL CORRETTO CARICAMENTO DEI PROGRAMMI SU CASSETTA PER COMMODORE 64

Scrivere LOAD o LOAD "nome programma" seguito dal tasto RETURN. Fare attenzione a digitare il nome in modo esatto.

Avviare il registratore con il tasto play ed attendere l'avvenuto caricamento del programma.

Se il programma non è provvisto di AUTOSTART (partenza automatica), digitare RUN seguito dal tasto RETURN.

Pur eseguendo le istruzioni sopraindicate è possibile incontrare talvolta qualche difficoltà nel caricamento del programma. La prima cosa da fare è assicurarsi che la testina del vostro registratore risulti ben pulita, allineata e smagnetizzata.

Non modificate per nessun motivo l'allineamento della testina del registratore poiché risulta pressoché impossibile, se non si dispone di apparecchiature professionali, procedere alla taratura dello stesso; in tal caso rivolgetevi presso un laboratorio specializzato.

PROGRAMMI CONTENUTI NELLA CASSETTA

Nella cassetta allegata sono presenti ordinatamente tutti i programmi proposti nelle 21 lezioni.

GARANZIA CASSETTE SOFTWARE

Le cassette software originali, non manomesse, che presentassero eventuali difetti vanno spedite per la sostituzione a:

EDIZIONI JCE

Via dei Lavoratori, 124

20092 Cinisello Balsamo (MI)

leader nell' elettronica

Ogni rivista JCE
 è leader indiscusso nel settore specifico,
 grazie alla ultra venticinquennale tradizione
 di serietà editoriale

SELEZIONE

DI ELETTRONICA E MICROCOMPUTER

È l'unica rivista italiana a carattere esclusivamente applicativo. Si rivolge ai progettisti di apparecchiature professionali, industriali e consumer. Col materiale che riceve dalle grandi Case, redige rubriche di alto interesse tecnologico dai titoli "Microprocessori" - "Microcomputer" - "Dentro al componente" - "Tecnologie avanzate". La rivista offre al lettore la possibilità di richiedere la documentazione.

MILLECANALI

È lo strumento critico che analizza e valuta obiettivamente l'emittenza radio e televisiva indipendente, quale elemento di rilievo nel cammino storico dei mezzi di informazione. Offre un valido supporto tecnico agli operatori, mantenendo il proprio ruolo nei confronti delle trasmissioni private e delle loro implicazioni nel contesto sociale.

SPERIMENTARE

CON L'ELETTRONICA E IL COMPUTER

La rivista, nata per gli hobbisti e affermata come periodico dei giovani, non ha mai abbandonato questa categoria di lettori. Sensibile all'evoluzione, si è arricchita della materia computer, divenendo una delle pubblicazioni leader nell'ambito dell'informatica di consumo. Contiene, fra l'altro, le rubriche "Sinclub" e "A tutto Commodore" che hanno avuto un ruolo determinante nel primato della rivista.

EG COMPUTER

È il mensile di home e personal computer. Pubblicazione unica nel suo genere, ricca di spunti entusiasmanti. È la rivista per il pubblico eterogeneo attratto dall'informatica, che intende varcarne le soglie in modo stimolante e vivace.

CINESCOPIO

Unica rivista italiana di Service Radiotelevisivo, per riparatori e operatori tecnici. Sempre aggiornata sulle nuove tecniche, offre un sostegno tangibile al Service-man nell'acquisizione di una più completa e moderna professionalità.



PUBBLICAZIONI JCE 1985

LIBRI DI INFORMATICA

| DESCRIZIONE | CODICE | PREZZO UNITARIO |
|---|--------|-----------------|
| IL LIBRO DEL MICRODRIVE SPECTRUM | 9001 | L. 16.000 |
| FORTH PER SPECTRUM | 9005 | L. 15.000 |
| ALLA SCOPERTA DEL QL IL COMPUTER SINCLAIR | 9050 | L. 20.000 |
| COME PROGRAMMARE IL TUO IBM PC | 9200 | L. 20.000 |
| LA PRIMA VOLTA CON APPLE | 9300 | L. 16.000 |
| ALLA SCOPERTA DELL'APPLE //c | 9301 | L. 16.000 |
| APPLE MACINTOSH: IL COMPUTER MAGICO | 9350 | L. 20.000 |
| ATARI SERIE XL | 9411 | L. 16.000 |
| IL 68000: PRINCIPI E PROGRAMMAZIONE | 9850 | L. 20.000 |
| | | |
| | | |

LIBRI DI INFORMATICA CON CASSETTA

| | | |
|--|------|-----------|
| SINCLAIR ZX SPECTRUM: ASSEMBLER E LINGUAGGIO MACCHINA | 9000 | L. 25.000 |
| PROGRAMMARE IMMEDIATAMENTE LO SPECTRUM | 9002 | L. 25.000 |
| CREATE GIOCHI ARCADE COL VOSTRO SPECTRUM | 9003 | L. 25.000 |
| APPROFONDIRE LA CONOSCENZA DELLO SPECTRUM | 9004 | L. 30.000 |
| PROGRAMMIAMO INSIEME LO SPECTRUM | 9006 | L. 30.000 |
| BASIC & FORTRAN PER SPECTRUM | 9007 | L. 25.000 |
| POTENZIATE IL VOSTRO SPECTRUM | 9008 | L. 30.000 |
| 49 GIOCHI ESPLOSIVI PER LO SPECTRUM | 9009 | L. 30.000 |
| GRAFICA AVANZATA CON LO SPECTRUM | 9010 | L. 35.000 |
| GRAFICA E SUONO PER IL LAVORO E IL GIOCO CON LO SPECTRUM | 9011 | L. 25.000 |
| METTETE AL LAVORO IL VOSTRO VIC 20 | 9100 | L. 25.000 |
| IL MIO COMMODORE 64 | 9150 | L. 25.000 |
| COME PROGRAMMARE IL TUO COMMODORE 64 | 9151 | L. 25.000 |
| COMMODORE 64: I SEGRETI DEL LINGUAGGIO MACCHINA | 9152 | L. 30.000 |
| SPRITES & SUONO DEL COMMODORE 64 | 9153 | L. 30.000 |
| SONY MSX BASIC | 9400 | L. 30.000 |
| IMPARIAMO IL PASCAL SUL NOSTRO COMPUTER | 9800 | L. 25.000 |
| | | |
| | | |

SOFTWARE

| | | | |
|---|---------------------|-----------|-----------|
| GRAFICA PER TUTTI | SPECTRUM 48k e PLUS | J/0100-01 | L. 25.000 |
| MANUALE DI GEOMETRIA PIANA | SPECTRUM 48k e PLUS | J/0100-02 | L. 25.000 |
| MANUALE DI GEOMETRIA SOLIDA | SPECTRUM 48k e PLUS | J/0100-03 | L. 25.000 |
| TRIGONOMETRIA | SPECTRUM 48k e PLUS | J/0100-04 | L. 25.000 |
| MOSAICO | SPECTRUM 48k e PLUS | J/0101-01 | L. 20.000 |
| BATTAGLIA NAVALE | SPECTRUM 48k e PLUS | J/0101-02 | L. 20.000 |
| PUZZLE MUSICALE | SPECTRUM 48k e PLUS | J/0101-03 | L. 20.000 |
| SUPER EG | SPECTRUM 48k e PLUS | J/0101-04 | L. 20.000 |
| SPECTRUM WRITER (MICRODRIVE COMPATIBILE) | SPECTRUM 48k e PLUS | J/0102-01 | L. 40.000 |
| MASTER FILE (MICRODRIVE COMPATIBILE) VERSIONE ITAL. | SPECTRUM 48k e PLUS | J/0102-02 | L. 40.000 |
| BUSINESS GRAPHICS | SPECTRUM 48k e PLUS | J/0102-03 | L. 25.000 |
| INGEGNERIA: PROGRAMMA AD ELEMENTI FINITI | SPECTRUM 48k e PLUS | J/0104-01 | L. 30.000 |
| TOPOGRAFIA | SPECTRUM 48k e PLUS | J/0104-02 | L. 30.000 |
| CALCOLO TRAVI IPE | SPECTRUM 48k e PLUS | J/0104-03 | L. 25.000 |
| ENERGIA SOLARE | SPECTRUM 48k e PLUS | J/0104-04 | L. 30.000 |
| ALGEBRA MATRICIALE | SPECTRUM 48k e PLUS | J/0104-05 | L. 30.000 |
| STUDIO DI FUNZIONI | SPECTRUM 48k e PLUS | J/0104-06 | L. 30.000 |

SOFTWARE

| DESCRIZIONE | CODICE | PREZZO UNITARIO |
|---|------------------------------|---------------------|
| EQUAZIONI PARAMETRICHE E PROBLEMI DI 2°GRADO | SPECTRUM 48k e PLUS | J/0104-07 L. 25.000 |
| TOTIP | SPECTRUM 48k e PLUS | J/0105-01 L. 20.000 |
| ASTROLOGIA | SPECTRUM 48k e PLUS | J/0105-02 L. 25.000 |
| CAMPIONATO DI CALCIO | SPECTRUM 48k e PLUS | J/0105-03 L. 25.000 |
| RACCOLTA DI QUIZ PER LA PATENTE (MICRODRIVE TRASFERIBILE) | SPECTRUM 48k e PLUS | J/0105-04 L. 25.000 |
| GARDEN WARS | COMMODORE C64 | J/0111-01 L. 20.000 |
| ECONOMIA FAMILIARE | COMMODORE C64/DISCO | J/0112-02 L. 40.000 |
| CHES WARS | COMMODORE VIC 20 NON ESPANSO | J/0121-01 L. 20.000 |
| | | |
| | | |

LIBRI DI ELETTRONICA

| | | |
|--|------|-----------|
| DIGIT 1 | 2000 | L. 7.000 |
| CORSO DI PROGETTAZIONE DEI CIRCUITI A SEMICONDOTTORE | 2002 | L. 8.400 |
| APPUNTI DI ELETTRONICA - VOL. 1 | 2300 | L. 8.000 |
| APPUNTI DI ELETTRONICA - VOL. 2 | 2301 | L. 8.000 |
| APPUNTI DI ELETTRONICA - VOL. 3 | 2302 | L. 8.000 |
| APPUNTI DI ELETTRONICA - VOL. 4 | 2303 | L. 8.000 |
| APPUNTI DI ELETTRONICA - VOL. 5 | 2304 | L. 8.000 |
| APPUNTI DI ELETTRONICA - VOL. 6 | 2305 | L. 8.000 |
| COSTRUIAMO UN VERO MICROELABORATORE ELETTRONICO | 3000 | L. 4.000 |
| JUNIOR COMPUTER - VOL. 1 | 3001 | L. 11.000 |
| JUNIOR COMPUTER - VOL. 2 | 3002 | L. 14.500 |
| GUIDA ALL'ACQUISTO DEI SEMICONDUTTORI | 4000 | L. 6.000 |
| TABELLA EQUIVALENZE SEMICOND. E TUBI ELETTRONICI PROFESSIONALI | 6006 | L. 5.000 |
| TRANSISTOR CROSS-REFERENCE GUIDE | 6007 | L. 8.000 |
| SELEZIONE DEI PROGETTI ELETTRONICI | 6008 | L. 9.000 |
| 300 CIRCUITI | 6009 | L. 12.500 |
| THE WORLD TTL, IC DATA CROSS REFERENCE GUIDE | 6010 | L. 20.000 |
| DIGIT 2 | 6011 | L. 6.000 |
| 273 CIRCUITI | 6014 | L. 12.500 |
| NUOVISSIMO MANUALE DI SOSTITUZIONE FRA TRANSISTORI | 6015 | L. 10.000 |
| SISTEMI HI-FI MODULARI DA 3 A 1000W | 6016 | L. 6.000 |
| 100 RIPARAZIONI TV ILLUSTRATE E COMMENTATE | 7000 | L. 10.000 |
| LE RADIO COMUNICAZIONI | 7001 | L. 7.500 |
| PRATICA TV | 7002 | L. 10.500 |
| 99 RIPARAZIONI TV ILLUSTRATE E COMMENTATE | 7003 | L. 16.000 |
| ALLA RICERCA DEI TESORI | 8001 | L. 6.000 |
| LE LUCI PSICHEDELICHE | 8002 | L. 4.500 |
| ACCESSORI ELETTRONICI PER AUTOVEICOLI | 8003 | L. 6.000 |
| IL MODERNO LABORATORIO ELETTRONICO | 8004 | L. 6.000 |
| LA PRATICA DELLE MISURE ELETTRONICHE | 8006 | L. 11.500 |
| | | |
| | | |

"ORE 10: LEZIONE DI BASIC" è un vero corso completo in 21 lezioni sul COMMODORE C 64 con il quale, con un'ora di applicazione al giorno, nel breve tempo di un mese, si può imparare la grammatica di base del più diffuso e popolare linguaggio per computer, il Basic. E' strutturato in una forma didattica particolarmente adatta ad introdurre l'informatica nelle scuole medie inferiori e superiori. E' di insostituibile aiuto anche a chi voglia apprendere da solo l'arte di dialogare col computer per riuscire a creare programmi semplici ma completi.

Il libro e la cassetta contengono 20 esercizi progressivi più un esercizio quadruplo finale composto da un indice e da 4 sottoprogrammi per la gestione completa dei dati con recupero da cassetta e stampa.

Ogni lezione contiene 5 sezioni: 1. Notizie di informatica 2. Esercitazione pratica 3. Fotografia dell'esecutivo (RUN) 4. Listato dell'esercizio con spiegazione 5. Programma in cassetta.

ISBN 88-7708-018-3

Cod. 9156

L. 29.000

LIBRO + CASSETTA

