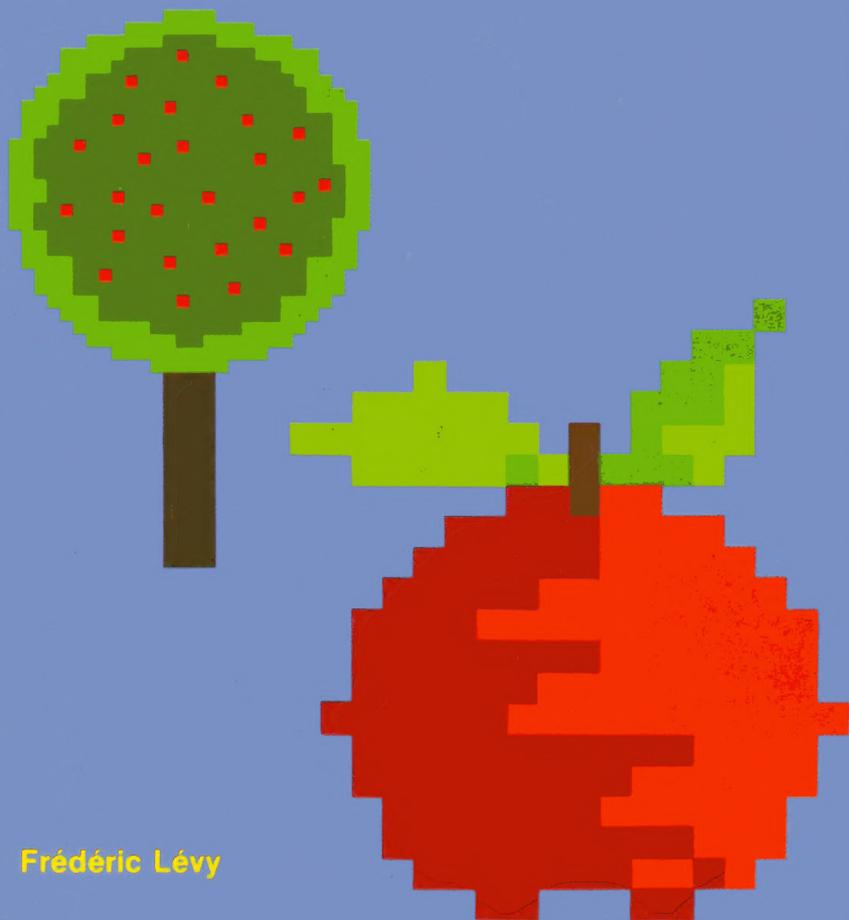
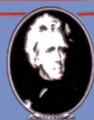


ALLA SCOPERTA DELL' APPLESOFT



Frédéric Lévy

EDIZIONE ITALIANA



GRUPPO
EDITORIALE
JACKSON

ALLA SCOPERTA DELL'APPLESOFT

**di
Frédéric Lévy**



**GRUPPO
EDITORIALE
JACKSON**
Via Rosellini, 12
20124 Milano

- © Copyright per l'edizione originale  Editions du P.S.I. 1981
© Copyright per l'edizione originale Gruppo Editoriale Jackson - 1984

Il Gruppo Editoriale Jackson ringrazia per il prezioso lavoro svolto nella stesura dell'edizione originale la signora Francesca Di Fiore e l'ing. Roberto Pancaldi.

Traduzione italiana a cura dell'ing. Enrico Odetti.

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

Fotocomposizione: Lineacomp S.r.l. - Via Rosellini, 12 - 20124 Milano

Stampato in Italia da:
S.p.A. Alberto Matarelli - Milano - Stabilimento Grafico

INTRODUZIONE

Questo libro è rivolto a tutte quelle persone che pur avendo già una conoscenza elementare del linguaggio BASIC desiderano però provare a scrivere alcuni semplici programmi con difficoltà crescente.

Il libro è diviso in due parti. Nella prima sono riportati gli enunciati dei problemi mentre nella seconda sono presentati i programmi risolutivi.

La prima parte è divisa a sua volta in sei capitoli ognuno dei quali rivolto ad un particolare argomento.

Gli enunciati sono in genere seguiti dall'analisi dettagliata del problema e qualche volta anche dal programma, quando esso rappresenti una nozione nuova e importante.

Per meglio utilizzare questo libro, vi consigliamo di provare a risolvere ogni problema voi stessi prima di leggere l'analisi che noi abbiamo proposto.

I programmi, che abbiamo riportato nella seconda parte, non sono ovviamente l'unica soluzione possibile, e proprio per questo motivo raccomandiamo al lettore di confrontarli, criticamente, con le proprie soluzioni.

PRESENTAZIONE DEGLI ESERCIZI

Gli enunciati dei problemi sono in genere formati da tre parti:

- *Problema: enunciazione dell'obiettivo del programma;*
- *Ingresso: descrizione dei dati che dovranno essere dati in ingresso dall'utente del programma (per esempio dei nominativi battuti alla tastiera);*
- *Uscita: indicazione dei dati che dovranno essere visualizzati sullo schermo del calcolatore.*

L'analisi comprende una descrizione del programma e un diagramma a blocchi.

Le soluzioni riportate nella seconda parte comprendono: il listato del programma, l'elenco delle variabili utilizzate e una spiegazione delle linee più importanti del programma.

SOMMARIO

PRIMA PARTE: ENUNCIATI E ANALISI

CAPITOLO 1 - VISUALIZZAZIONI, CALCOLI SEMPLICI, GENERAZIONE DI NUMERI ALEATORI	1
CAPITOLO 2 - RICERCHE IN TABELLE, CALCOLI, ARCHIVIAZIONI, INSERIMENTI	7
CAPITOLO 3 - RICERCHE DENTRO LISTE DI CARATTERI, INSERIMENTI	21
CAPITOLO 4 - ISTOGRAMMI, GRAFICI IN ALTA E BASSA RISOLUZIONE, TRACCIAMENTO DI FUNZIONI, ANIMAZIONI	33
CAPITOLO 5 - GIOCHI GRAFICI, "FILETTO", ESTRAZIONE DI UNA SERIE DI NUMERI ALEATORI, NUMERI PRIMI	47
CAPITOLO 6 - SCRITTURA E LETTURA DI DATI IN UN ARCHIVIO, PICCOLO PROGRAMMA DI GESTIONE INDIRIZZI	55
SECONDA PARTE: PROGRAMMI	61

PRIMA PARTE

ENUNCIATI E ANALISI

CAPITOLO 1

VISUALIZZAZIONI, CALCOLI SEMPLICI, GENERAZIONE DI NUMERI ALEATORI

ESEMPIO 1.1

Problema: visualizzazione delle iniziali e della lunghezza del cognome di una persona.

Ingresso: cognome e nome della persona.

Uscita: le iniziali e la lunghezza del cognome.

Soluzione:

```
10 HOME : VTAB 5
20 INPUT "COGNOME: "; C$
30 INPUT "NOME: "; N$
40 PRINT : PRINT
50 HTAB 10
60 PRINT LEFT$ (N$,1); ". "; LEFT$
   (C$,1); ". "
70 PRINT
80 PRINT "LUNGHEZZA DEL COGNOME:
   "; LEN (C$)
90 END
```

Variabili:

C\$ - cognome

N\$ - nome

60: visualizzazione delle iniziali, ciascuna seguita da un punto.

ESERCIZIO 1.2

Problema: centrare un titolo sullo schermo.

Ingresso: un titolo (o una frase) con meno di 40 caratteri.

Uscita: il titolo deve apparire a metà dello schermo (sulla linea 10).

ESERCIZIO 1.3

Ingresso: cognome e anno di nascita di una persona.

Uscita: visualizzare i testi:

BUON GIORNO (cognome)

NON DIRO' A NESSUNO CHE HAI (età) ANNI

Analisi:

Il programma deve:

- 1) domandare il cognome,
- 2) domandare l'anno di nascita,
- 3) visualizzare i testi indicati sopra con l'età.

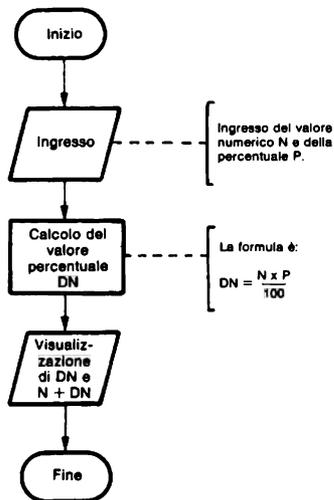
ESERCIZIO 1.4

Problema: calcolo della percentuale di un valore numerico n.

Ingresso: il valore numerico e la percentuale.

Uscita: il valore maggiorato per la percentuale.

Analisi:

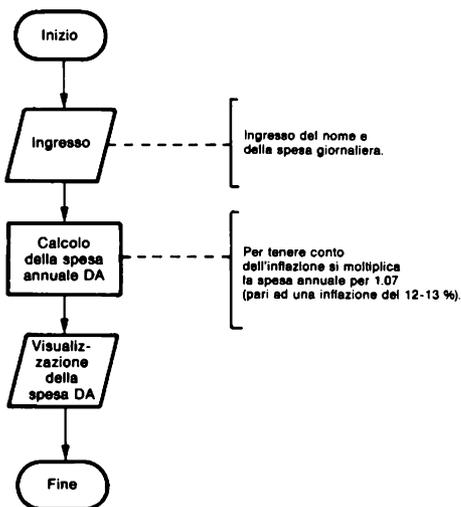


ESERCIZIO 1.5

Problema: calcolo approssimativo della spesa annua a partire dal valore medio giornaliero e tenendo conto dell'inflazione.

Ingresso: valore della spesa media giornaliera.
Uscita: spesa annuale (usare le istruzioni HOME, VTAB, ecc. per rendere chiara la visualizzazione sullo schermo).

Analisi:



ESEMPIO 1.6

Problema: simulazione del lancio di un dado.

Uscita: un numero intero aleatorio compreso tra 1 e 6.

Analisi: La funzione RND(1) genera un numero aleatorio tra 0 compreso e 1 escluso; moltiplicando per 6 si ottengono valori tra 0 compreso e 6 escluso; si prende allora la parte intera, dopo aver aggiunto 1, per ottenere i valori interi tra 1 e 6, estremi inclusi. In BASIC : $INT(RND(1)*6)+1$.

Soluzione:

```
10 HOME : VTAB 5
20 D = RND (1) * 6 + 1
30 D = INT (D)
40 PRINT D
50 PRINT
60 INPUT "UN ALTRO LANCIO DI DAD
   0? <SI/NO>"; A$
70 PRINT
80 IF A$ = "SI" THEN 20
90 END
```

Variabili:

D - numero intero aleatorio tra 1 e 6

A\$ - variabile

20-30: calcolo del valore numerico aleatorio.

60: richiesta se continuare il lancio del dado (tra parentesi sono riportate le due risposte possibili).

80: se l'utente risponde sì il programma prosegue alla linea 20.

ESERCIZIO 1.7

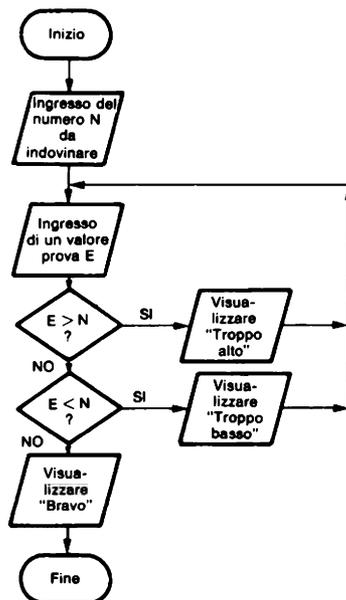
Problema: gioco; l'utilizzatore deve scoprire, con tentativi successivi, un numero stabilito da un'altra persona.

Ingresso: il numero da scoprire (compreso tra 0 e 1000) e i valori tentativi.

Uscita: dare i messaggi: "TROPPO ALTO", "TROPPO BASSO", "BRAVO INDOVINATO".

Analisi:

Dopo ogni tentativo il calcolatore deve controllare se il numero fornito è superiore o inferiore a quello da indovinare e poi visualizza il messaggio corrispondente; diversamente annuncia che il numero è stato indovinato.



ESERCIZIO 1.8

Ingresso: numero di dadi gettati assieme (da 1 a 6).

Uscita: la somma dei dadi gettati.

ESERCIZIO 1.9

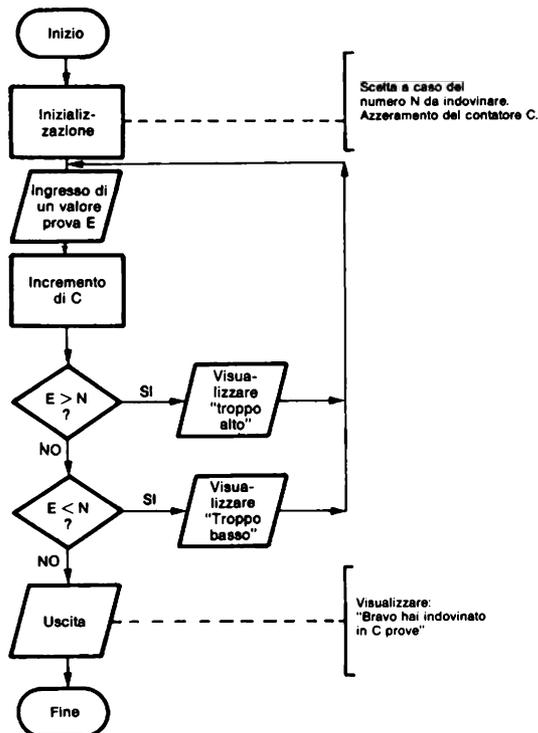
Problema: come la versione 1.7, ma migliorata perchè è la macchina che stabilisce, a caso, il numero da indovinare e conta in quante volte l'operazione ha successo.

Ingresso: i valori numerici tentativi.

Uscita: gli stessi messaggi della versione 1.7 con in più il numero di tentativi.

Analisi:

La variabile C memorizza il numero di tentativi e per questo motivo viene azzerata all'inizio del programma; C viene quindi "incrementata" di 1 ad ogni tentativo. (La variabile C viene chiamata "contatore").



CAPITOLO 2

RICERCHE IN TABELLE, CALCOLI, ARCHIVIAZIONI, INSERIMENTI

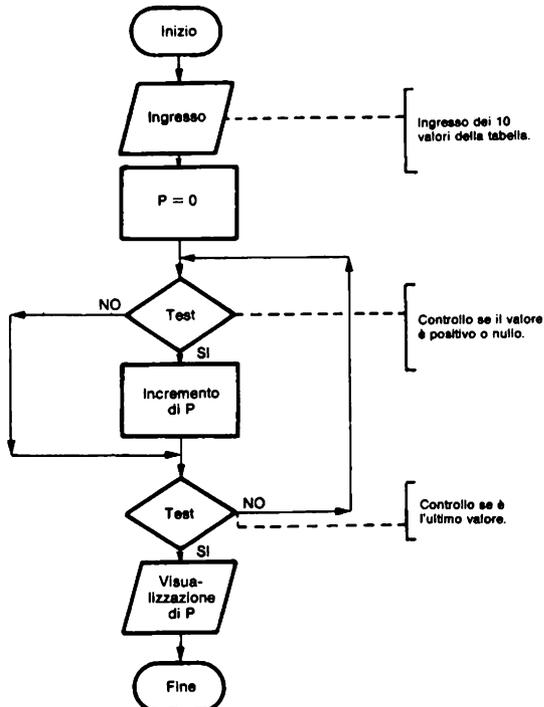
ESEMPIO 2.1

Problema: trovare il numero di valori positivi o nulli contenuti in una lista di dieci elementi.

Ingresso: dieci valori numerici.

Uscita: il numero di valori positivi o nulli.

Analisi:



È necessario usare un contatore (la variabile P) che sarà incrementato di 1 per ogni valore positivo o nullo.

I dati da analizzare vengono inizialmente caricati in una tabella.

Soluzione:

```
10 DIM V(10)
20 HOME : VTAB 5
25 :
30 REM SCELTA DEI VALORI
40 FOR I = 1 TO 10
50 PRINT "VALORE "; I:
60 INPUT " : "; V(I)
70 NEXT I
75 :
80 REM VALORI POSITIVI O NULLI?

90 P = 0
100 FOR I = 1 TO 10
110 IF V(I) >= 0 THEN P = P +
    1
120 NEXT I
125 :
130 REM VISUALIZZAZIONE
140 PRINT : PRINT
150 PRINT "NUMERO DI VALORI POSI
    TIVI: "; P
160 END
```

Variabili:

I - contatore

P - contatore dei valori positivi o nulli

10: dimensionamento della tabella.

90: inizializzazione del contatore P.

ESERCIZIO 2.2

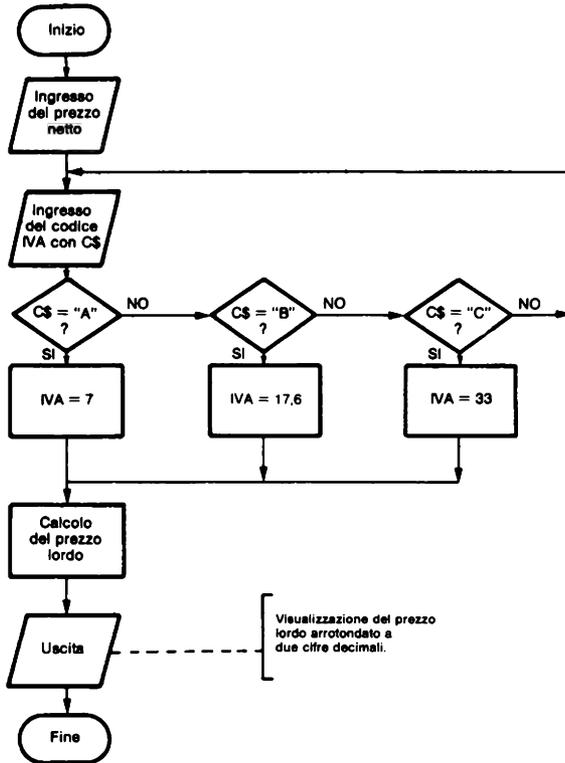
Problema: calcolo di un prezzo lordo a partire da quello al netto dell'IVA.

Ingresso: prezzo netto e codice del valore IVA (A per IVA del 7%, B per IVA del 17.6%, C per IVA del 33%).

Uscita: il prezzo lordo approssimato a due cifre decimali.

Analisi:

La percentuale IVA viene associata ad una variabile numerica mediante la quale si calcola poi il prezzo lordo. Se viene fornito un codice IVA errato, il programma chiede la ripetizione del codice.



ESERCIZIO 2.3

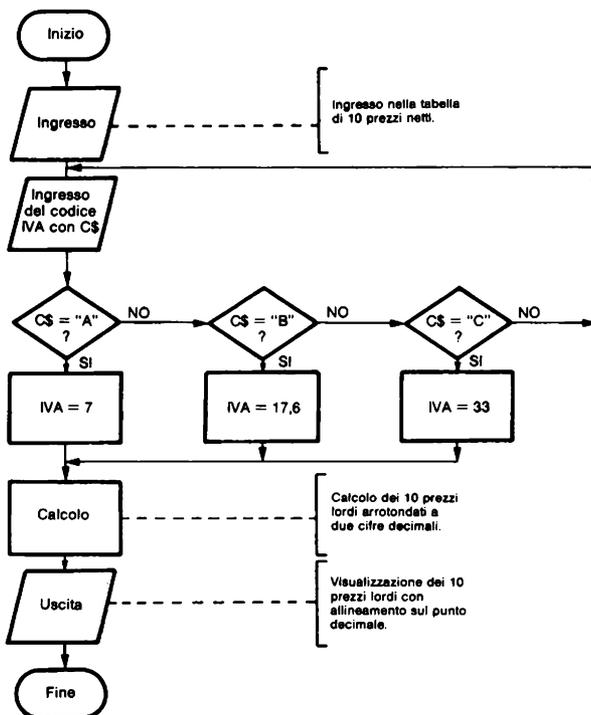
Problema: come il precedente Esercizio 2.2, ma per un gruppo di dieci prezzi alla volta.

Ingresso: dieci prezzi netti e il codice IVA.

Uscita: i dieci prezzi lordi approssimati a due cifre decimali e allineati a destra sul punto decimale.

Analisi:

L'utilizzo di una tabella facilita la registrazione dei dieci prezzi e dei calcoli. Si usano per questo scopo le istruzioni FOR-NEXT.



ESERCIZIO 2.4

Problema: lettura di una lista di parole registrate nel programma mediante l'istruzione DATA (trovare due metodi diversi per la risoluzione del problema).

Ingresso: la lista di parole nell'istruzione DATA.

Uscita: la visualizzazione delle parole.

ESERCIZIO 2.5

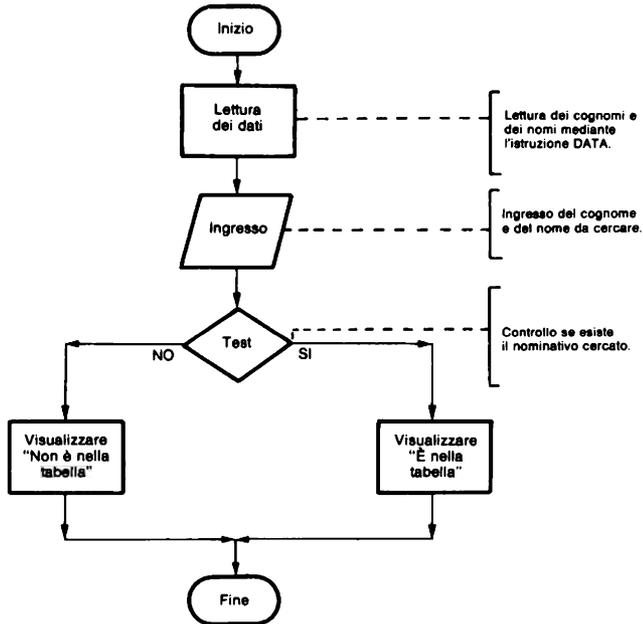
Problema: vedere se un nominativo è presente in una tabella di nomi registrata nel programma.

Ingresso: una lista di cognomi e nomi e il cognome e nome della persona da cercare.

Uscita: un messaggio per dire se il nominativo è contenuto nella tabella.

Analisi:

Si utilizza l'istruzione DATA per costituire la tabella.
La ricerca si effettua comparando uno per uno i cognomi e i nomi con quello voluto.



ESERCIZIO 2.6

Problema: trovare il valore più piccolo e quello più grande in una lista di numeri.

Ingresso: una lista di numeri.

Uscita: il valore più grande e quello più piccolo.

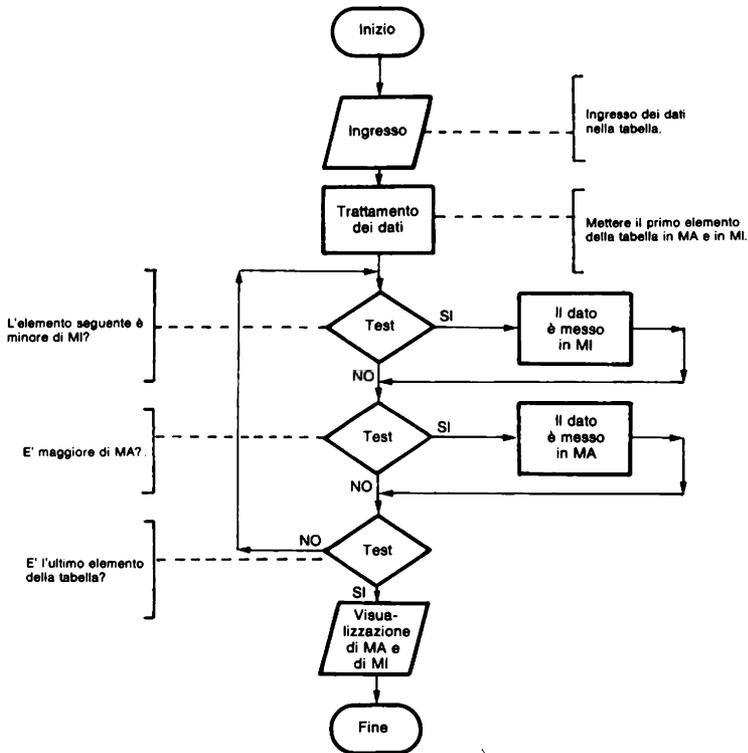
Analisi:

Il valore maggiore si trova nel modo seguente:

- mettere il primo elemento della lista nella variabile MG,
- comparare uno ad uno gli elementi della lista con MG. Se uno è maggiore di MG allora è lui che viene posto in MG, diversamente MG non viene cambiato.

Al termine del confronto di tutti gli elementi, MG contiene il valore maggiore della lista.

Per trovare invece il valore minore la procedura è la stessa salvo mettere nella nuova variabile MP l'elemento inferiore dei confronti.



ESERCIZIO 2.7

Problema: trovare l'indice della prima persona in ordine alfabetico, in una lista di nominativi.

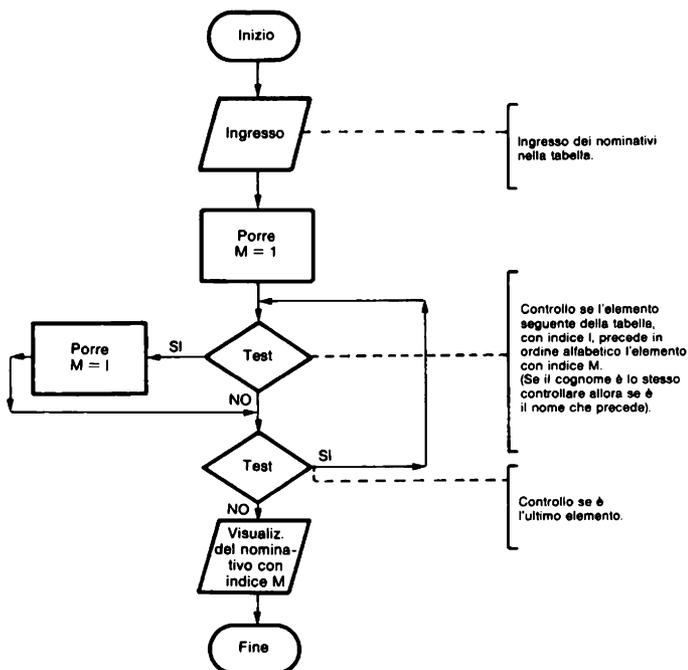
Ingresso: una lista di cognomi e nomi.

Uscita: l'indice della prima persona in ordine alfabetico e i corrispondenti nome e cognome.

Analisi:

Il metodo per effettuare la ricerca è lo stesso dell'esercizio precedente salvo conservare l'indice dell'elemento più piccolo, in senso alfabetico.

E' inoltre necessario fare due gruppi di confronti nel caso si trovino più persone con lo stesso cognome.



ESERCIZIO 2.8

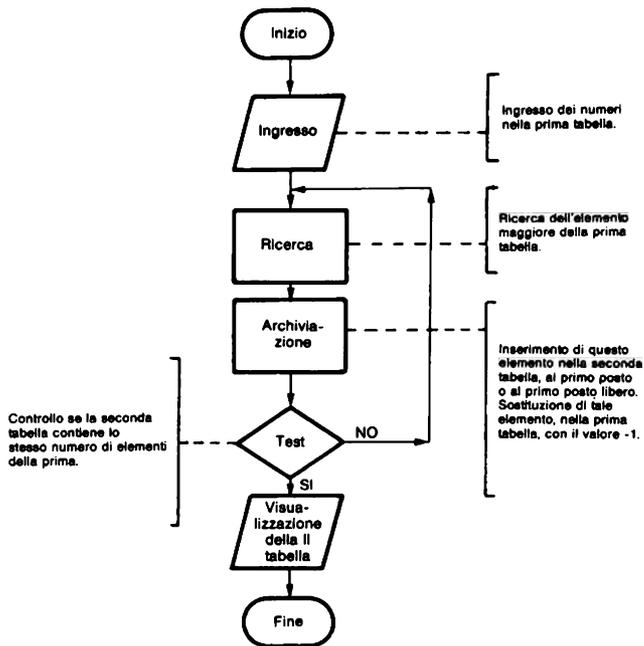
Problema: archiviare per ordine decrescente una lista di numeri forniti al programma con ordine qualunque, utilizzando due tabelle: la prima per contenere i numeri da ordinare e la seconda per fare l'ordinamento.

Ingresso: una lista di numeri.

Uscita: una lista ordinata in verso decrescente.

Analisi:

Proponiamo due metodi diversi:

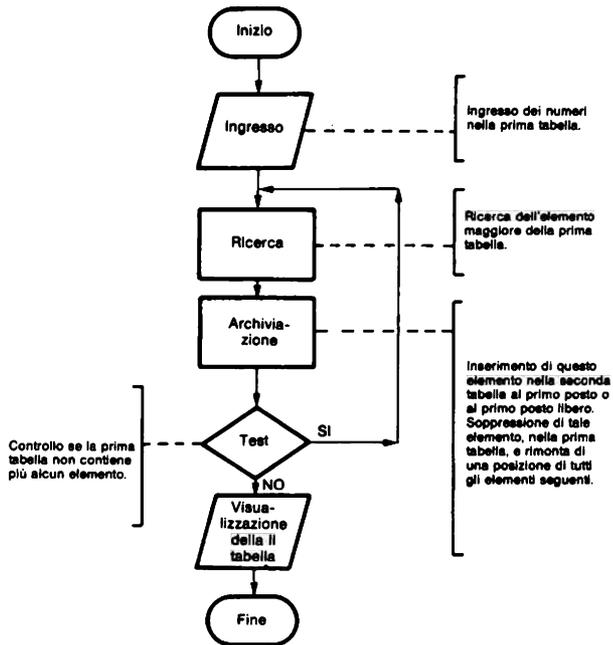


Metodo 1

È necessario scegliere un numero particolare che non possa mai essere presente nella lista.

Nel nostro caso stabiliamo che i numeri siano tutti positivi o nulli per cui scegliamo -1 come numero speciale che essendo negativo non potrà, quindi, mai essere presente.

Il calcolatore ricerca il valore maggiore nella prima tabella e poi lo pone nella seconda tabella; nella prima tabella al suo posto scrive -1. Questo ciclo viene ripetuto sino a che nella prima tabella siano presenti solo valori -1. Alla fine nella seconda tabella saranno presenti tutti i numeri in ordine decrescente.



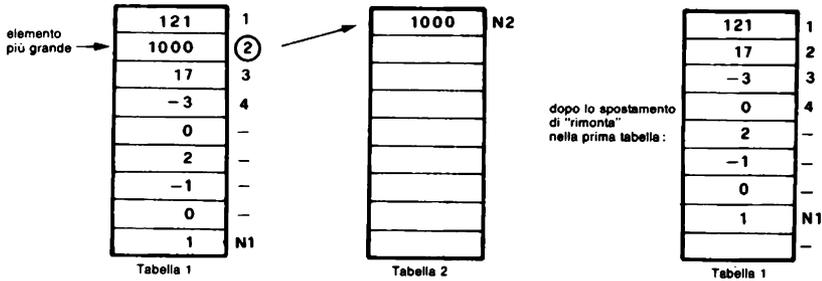
Metodo 2

Questo metodo è applicabile in modo più generale qualunque siano i numeri da ordinare perchè non richiede la scelta di un valore speciale.

Appena avete trovato l'elemento più grande della prima tabella, ed averlo trasferito nella seconda, esso viene soppresso facendo "rimontare" tutti gli elementi successivi di una posizione. La prima tabella viene quindi ad avere, dopo questa operazione, un elemento in meno.

Il procedimento si ripete sino a che non vi sono più elementi nella prima tabella.

Esempio :



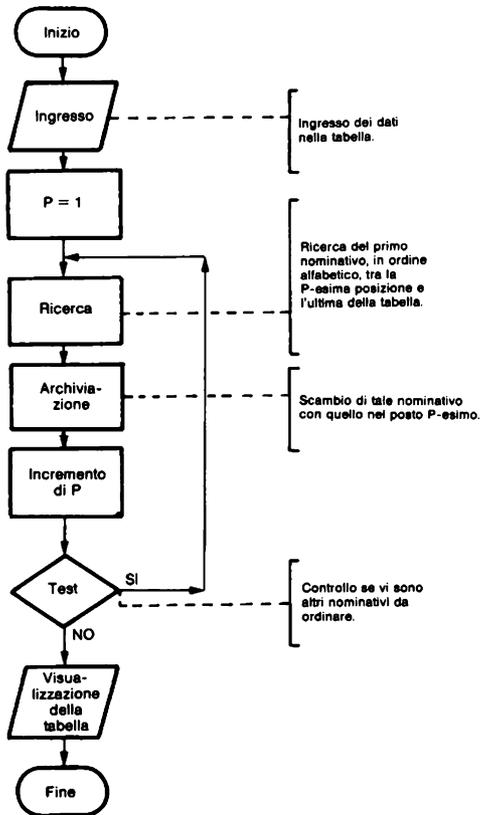
Il primo metodo ha il vantaggio di essere più semplice, ma richiede la definizione di un valore speciale che non deve mai essere contenuto alla tabella iniziale (-1).

Il secondo metodo si applica a qualunque tipo di tabella, ma è più complicato e più lungo da eseguire a causa dei tanti spostamenti di rimonta.

ESERCIZIO 2.9

- Problema:** come l'Esercizio 2.8 precedente, ma per una lista di nominativi da ordinare alfabeticamente (utilizzare una sola tabella).
- Ingresso:** una lista di nominativi.
- Uscita:** la stessa lista ordinata alfabeticamente.

Analisi:



Per risolvere problemi di questo tipo è necessario utilizzare una sola tabella perchè la memoria del calcolatore potrebbe non essere sufficiente per contenere più tabelle.

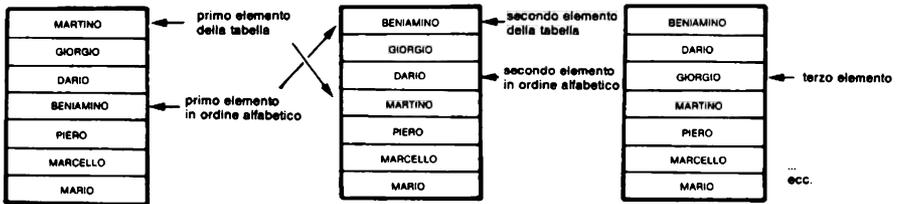
Il metodo che utilizziamo (applicato ad una tabella di soli cognomi) consiste nel ricercare il cognome più piccolo, cioè il primo in ordine alfabetico, e scambiarlo con il primo elemento della tabella.

In tal modo il primo elemento della tabella diviene il primo cognome in ordine alfabetico.

Si prosegue poi analizzando tutti gli elementi successivi spostando il più piccolo, dei rimanenti, nella seconda posizione, poi nella terza e così via.

Per maggiore chiarezza osservate l'esempio seguente.

Esempio:



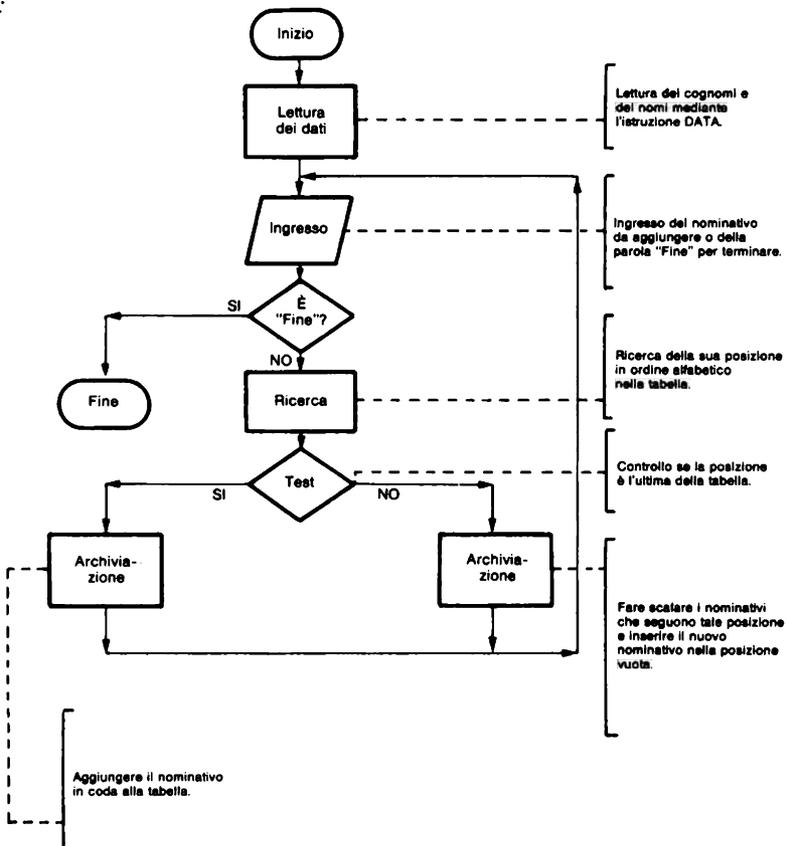
ESERCIZIO 2.10

Problema: aggiungere un nominativo ad una tabella già ordinata alfabeticamente, nella posizione corretta in ordine alfabetico.

Ingresso: una lista di nomenclativi (cognomi e nomi) e il nuovo nominativo.

Uscita:

Analisi:

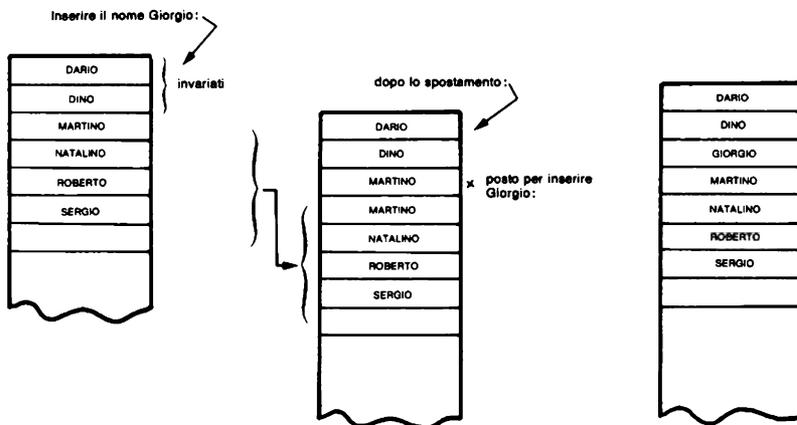


Per inserire il nuovo nominativo è necessario dapprima trovare la sua posizione nella tabella.

Se questa posizione è alla fine della tabella, è allora sufficiente aggiungere il nome in coda alla tabella. Diversamente è necessario spostare di una posizione tutti i nominativi successivi alla posizione trovata.

Anche in questo caso l'esempio seguente vi sarà di chiarimento.

Esempio:



RICERCHE DENTRO LISTE DI CARATTERI, INSERIMENTI, SOPPRESSIONE DI CARATTERI

ESERCIZIO 3.1

Problema: sopprimere i caratteri “spazio” che si trovano in coda ad una lista di caratteri. (Talvolta è necessario che siano tagliati i caratteri spazio posti all’inizio e in coda ad una stringa. Questo esercizio mostra come togliere quelli in coda; quelli all’inizio vengono tolti automaticamente.)

Ingresso: una linea di caratteri.

Uscita: la stessa linea privata però dei caratteri spazio posti in coda.

ESERCIZIO 3.2

Problema: sopprimere tutti i caratteri “spazio” che si trovano in una stringa.

Ingresso: una linea di caratteri.

Uscita: la stessa linea senza spaziature.

ESERCIZIO 3.3

Problema: aggiungere uno spazio tra i caratteri di una stringa.

Ingresso: una linea di caratteri.

Uscita: la stessa linea con spaziature tra i caratteri.

ESERCIZIO 3.4

Ingresso: una frase o una parola.

Uscita: la stessa frase o parola, ma con una lettera in meno ogni volta, come mostrato nell’ esempio seguente.

Esempio: SALUTI diviene
 SALUTI
 ALUTI
 LUTI
 UTI
 TI
 I

ESERCIZIO 3.5

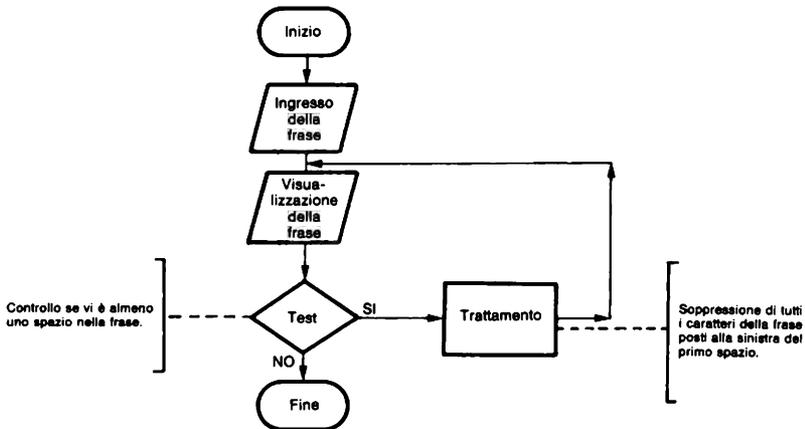
Ingresso: una frase.

Uscita: visualizzazioni successive della stessa frase, ma con una parola in meno ogni volta come mostrato nell'esempio.

Esempio: TRE CIVETTE SUL COMO'
 CIVETTE SUL COMO'
 SUL COMO'
 COMO'

Analisi:

Supponiamo, ovviamente, che tutte le parole della frase siano separate da un carattere "spazio". Il programma ricerca appunto il primo spazio da sinistra e sopprime la parola subito precedente. Visualizza quindi la nuova frase e ricomincia la ricerca del carattere spazio.



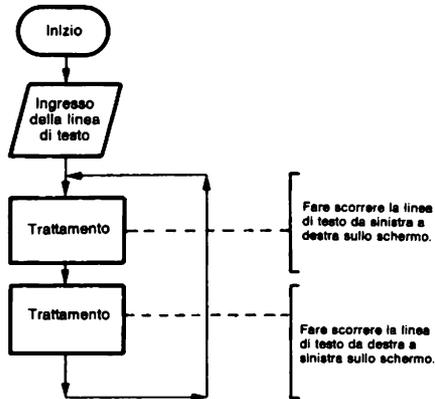
ESEMPIO 3.6

Problema: fare scorrere una frase sullo schermo da sinistra a destra e da destra a sinistra continuamente.

Ingresso: una frase con meno di 20 caratteri.

Uscita: la stessa frase che scorre sullo schermo.

Analisi:



Soluzione:

```
10 HOME : VTAB 5
20 INPUT "LINEA (20 CAR MAX) : "
   ;L$
30 L = LEN (L$) : IF L > 20 THEN
   20
40 REM DA SINISTRA A DESTRA
50 FOR I = 1 TO 40 - L
60 VTAB 15: PRINT SPC( I);L$
70 FOR J = 1 TO 100: NEXT J
80 NEXT I
90 REM DA DESTRA A SINISTRA
100 FOR I = 40 - L TO 0 STEP -
   1
110 VTAB 15: PRINT SPC( I);L$. SPC(
   40) : PRINT
120 FOR J = 1 TO 100: NEXT J
130 NEXT I
140 GOTO 40
```

Variabili:

- I,J - contatori
- L - lunghezza di L\$
- L\$ - linea di testo

- 50, 100 : la linea viene visualizzata dalla colonna 0 alla colonna 40-L dove L è la lunghezza della linea di testo. (L'istruzione PRINT in coda alla linea 110 serve per ritornare a capo).
- 60, 110: la linea di testo viene fatta precedere o seguire da spaziature per cancellare la visualizzazione precedente.
- 70, 120: ciclo di attesa.

ESERCIZIO 3.7

Problema: fare scorrere una frase sullo schermo da destra a sinistra senza mai fermarsi.

Ingresso: una frase con meno di 40 caratteri.

Uscita: la stessa frase che scorre sullo schermo.

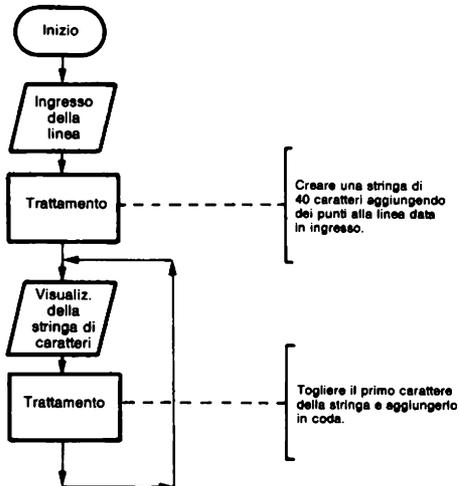
Analisi:

Proponiamo due metodi per risolvere il problema:

Metodo 1

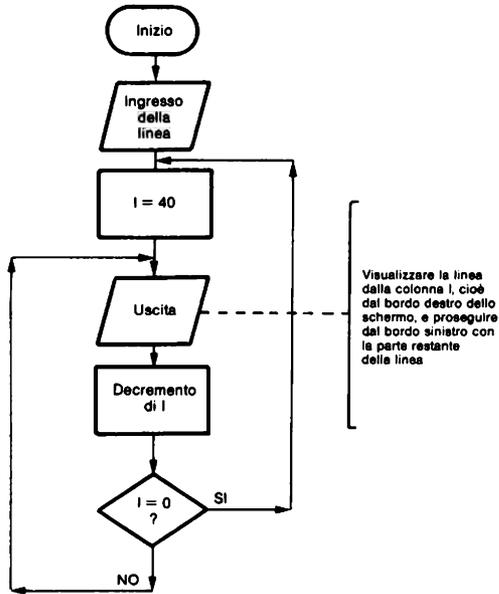
Creare una stringa di 40 caratteri (larghezza dello schermo) composta dalla frase d'ingresso e da un numero sufficiente di caratteri spazio e poi visualizzarla sullo schermo. Per farla scorrere elidere il suo primo carattere e aggiungerlo in coda.

Proseguire quindi con visualizzazioni successive.



Metodo 2

Questo metodo prevede l'uso dell'istruzione MID\$.



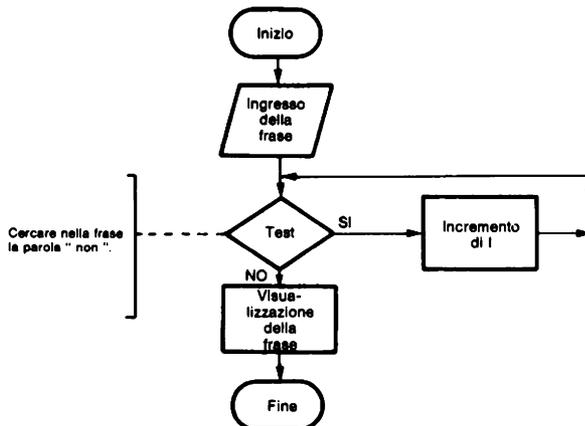
ESERCIZIO 3.8

Problema: trasformare una frase dalla forma negativa a quella positiva sopprimendo la parola NON.

Ingresso: una frase con significato negativo.

Uscita: la stessa frase, ma con significato affermativo.

Analisi:



Attenzione: la ricerca delle negazioni deve essere fatta cercando la parola “NON ” preceduta e seguita da spaziature per evitare confusione con parole come “NON-no”, “anNONario”, ecc. Attenzione anche al caso che la negazione sia all’inizio della frase.

ESERCIZIO 3.9

Problema: inserire alcuni caratteri in una stringa in una posizione voluta.

Ingresso: una stringa di caratteri, i caratteri da inserire e la loro posizione.

Uscita: la nuova stringa.

Esempio: “APPLE II” e “PLUS ” con 6 danno “APPLE PLUS II”.

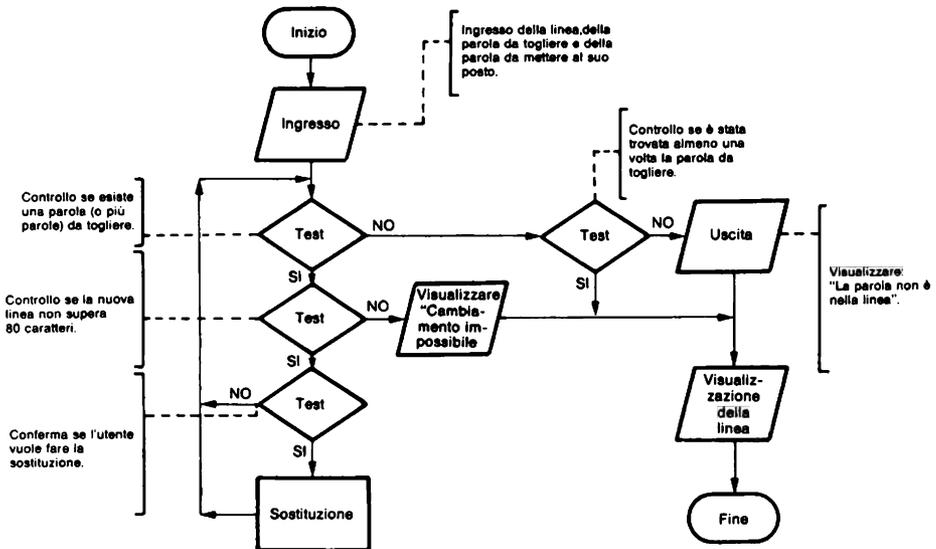
ESERCIZIO 3.10

Problema: sostituzione di una parola con un’altra in una linea di testo non più lunga di 80 caratteri. Se la parola è presente più volte, deve essere sempre sostituita (purchè non venga superata la lunghezza massima di 80 caratteri).

Ingresso: una linea di testo non più lunga di 80 caratteri, la parola da togliere e quella nuova da mettere.

Uscita: la nuova linea (non più lunga di 80 caratteri).

Analisi:



La ricerca della posizione della parola da cambiare nella frase, viene effettuata confrontandola con un gruppo contiguo di caratteri della stessa lunghezza, comin-

ciando dall'inizio e proseguendo con spostamenti successivi verso destra di un carattere sino alla fine della linea.

Ogni volta che viene incontrata la parola da togliere, il calcolatore chiede conferma della sostituzione all'utente.

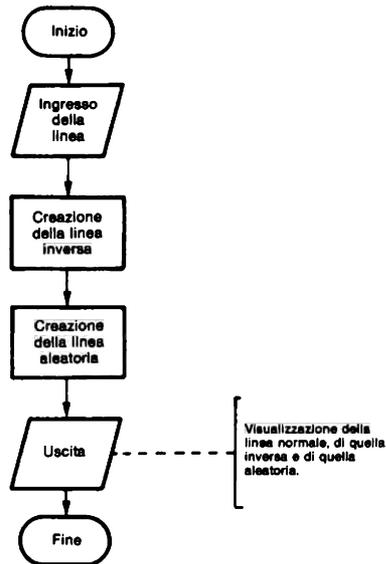
ESERCIZIO 3.11

Ingresso: una frase o una parola.

Uscita: la stessa frase con i caratteri tutti invertiti e un'altra frase con gli stessi caratteri, ma posti in ordine aleatorio.

Esempio: TAVOLA da ALOVAT in ordine inverso e ALVOTA o VLAATO, per esempio, in ordine qualunque.

Analisi:



Per invertire i caratteri si proceda nel modo seguente: si leggano uno ad uno dal fondo i caratteri della stringa e li si riscrivano in una nuova stringa.

Esempio: PINOCCHIO diviene

O
OI
OIH
OIHC
OIHCC
OIHCCO
OIHCCON
OIHCCONI
OIHCCONIP

Per costruire invece la stringa aleatoria, si preleva uno alla volta a caso un carattere della stringa originaria, cancellandolo da questa, e lo si riscrive nella nuova stringa.

Esempio: PINOCCHIO diviene

	PNOCCHIOI
	PNOCHIOIC
	NOCHIOICP
	NOCHOICPI
	NOCHICPIO
	OCHICPION
	OCICPIONH
	CICPIONHO
	ICPIONHOC

ESERCIZIO 3.12

- Problema:** visualizzare il codice ASCII di un carattere battuto alla tastiera (utilizzare l'istruzione GET).
- Ingresso:** un carattere battuto alla tastiera.
- Uscita:** il codice ASCII del carattere battuto (con 'Control C' cioè codice 3 il programma termina).

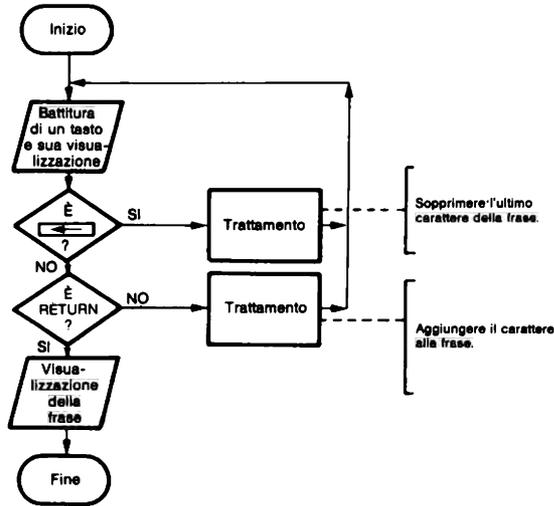
Soluzione:

```
10 GET A$
20 PRINT ASC (A$)
30 IF ASC (A$) < > 3 THEN 10
40 END
```

ESERCIZIO 3.13

- Problema:** l'istruzione INPUT non permette che siano dati in ingresso in una stringa ne le virgole ne i due punti. Provate a simulare tale istruzione con la GET così da potere accettare anche questi due caratteri in ingresso.
- Ingresso:** una linea di testo che comprenda anche virgole e due punti.
- Uscita:** la stessa linea di testo.

Analisi:



Attenzione: l'istruzione GET A\$ attende che venga battuto un qualunque tasto e poi attribuisce, alla variabile A\$, il carattere corrispondente. RETURN e sono per esempio due caratteri ASCII. Per sapere quale dei due valori è contenuto nella variabile A\$ si può usare la funzione ASC:

Codice ASCII di RETURN : 13
di : 8

ESERCIZIO 3.14

Problema: trasformare ciascun elemento, di una tabella di numeri, nella sua percentuale in rapporto alla somma di tutti i numeri della tabella.

Ingresso: una tabella di numeri.

Uscita: la tabella delle singole percentuali calcolate rispetto alla somma di tutti i valori della tabella.

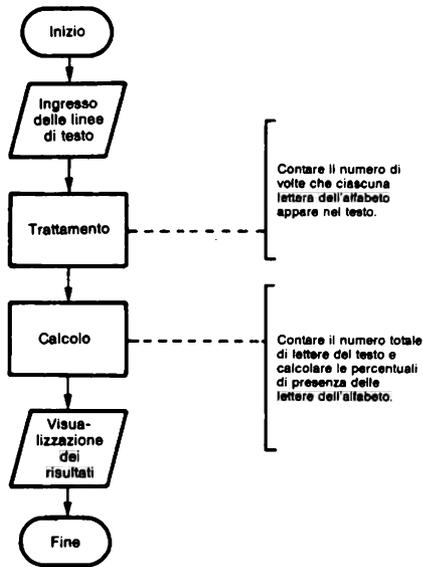
ESERCIZIO 3.15

Problema: visualizzare la percentuale di presenza in un testo di ciascuna lettera dell'alfabeto.

Ingresso: una o più linee di un testo.

Uscita: tutte le lettere dell'alfabeto seguite dalle percentuali di presenza nel testo dato.

Analisi:



ESERCIZIO 3.16

Problema: trasformare una data, posta nella forma numerica 15,7,1790 (giorno, mese, anno), in una stringa di caratteri "15 luglio 1790"

Ingresso: tre numeri corrispondenti a giorno, mese e anno.

Uscita: visualizzazione della stringa contenente la data.

ESERCIZIO 3.17

Problema: programma inverso del precedente (Esercizio 3.16).

Ingresso: una data sotto forma di stringa.

Esempio: 13 luglio 1788

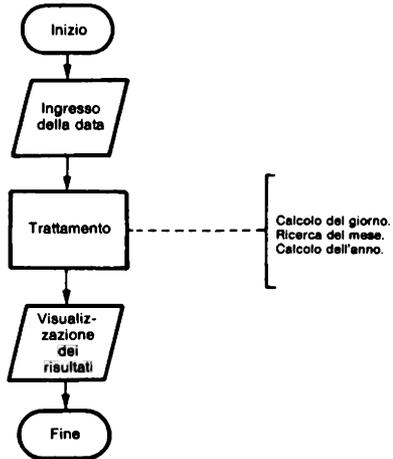
Uscita: i tre numeri corrispondenti a giorno, mese e anno.

Esempio: 13, 7, 1788

Analisi:

La stringa originaria viene scomposta in base alle spaziature.

Per ricercare il numero del mese, i mesi possono essere registrati in ordine da gennaio a dicembre in una istruzione DATA.



ISTOGRAMMI, GRAFICI IN ALTA E BASSA RISOLUZIONE, TRACCIAMENTO DI FUNZIONI, ANIMAZIONI

ESEMPIO 4.1

Problema: sottolineare il cognome e il nome di una persona.

Ingresso: il cognome e il nome di una persona.

Uscita: il cognome e il nome sottolineati.

Soluzione:

```

10 TR$ = "-----"
   "-----"
20 HOME : VTAB 5
30 INPUT "NOME? "; N$
40 INPUT "COGNOME? "; C$
50 PRINT : PRINT
60 HTAB 5
70 PRINT N$; " "; C$
80 HTAB 5
90 PRINT MID$(TR$,1, LEN(N$))
   ;" "; MID$(TR$,1, LEN(C$))
   )
100 END
    
```

Variabili:

N\$ - nome

C\$ - cognome

Attenzione: è possibile scrivere:

```
90 PRINT LEFT$(TR$,LEN(N$));" ";LEFT$(TR$,LEN(C$))
```

ma questa linea di programma può causare un errore se N\$ o C\$ sono nulle.

ESERCIZIO 4.2

Problema: visualizzare sullo schermo un linea e una colonna di asterischi.

Ingresso: un numero L compreso tra 1 e 24, e un numero C compreso tra 1 e 39.

Uscita: visualizzare una linea di C asterischi sulla L-esima linea e una colonna di L asterischi sulla C-esima colonna.

ESERCIZIO 4.3

Problema: visualizzare un istogramma orizzontale a partire da venti valori.

Ingresso: venti valori compresi tra 0 e 100.

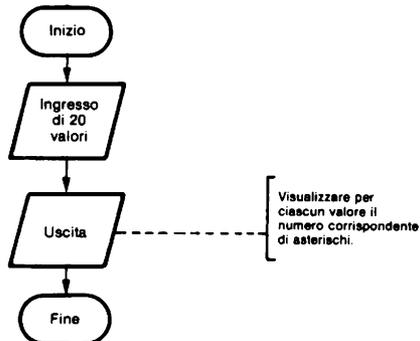
Uscita: visualizzazione dell'istogramma corrispondente.

Esempio: 10,35,90,3,8,... danno luogo all'istogramma:

```
####  
#####  
#####  
#  
###
```

Analisi:

Si fa corrispondere il valore massimo (per noi 100) al massimo numero di asterischi che si possono tracciare su una riga dello schermo (40). I valori intermedi dell'istogramma sono ottenuti con una semplice proporzione che per un valore generico V è: $V/100*40$



Esercizio 4.4

Problema: analogo all'Esercizio 4.3 precedente, ma per 30 valori positivi senza limitazione di grandezza e con l'istogramma posto verticalmente.

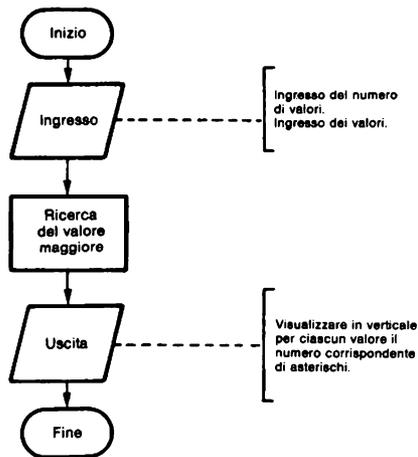
Ingresso: 30 valori positivi.

Uscita: l'istogramma verticale corrispondente.

Analisi:

In questo caso il numero massimo di asterischi, visualizzabili su una colonna (24), viene fatto corrispondere al valore maggiore dato in ingresso.

La formula che ci dà il numero di asterischi, a partire dal valore massimo G e dal valore corrente V, è: $V/G*24$



ESEMPIO 4.5

Ingresso: un numero compreso tra 0 e 15.

Uscita: riempire lo schermo, nel modo grafico a bassa risoluzione, con il colore il cui codice è richiesto in ingresso.

Soluzione:

```
10 TEXT : HOME : VTAB 5
20 INPUT "QUALE COLORE (0-15) ?
   ": C
30 IF C < 0 OR C > 15 THEN 20
40 GR
50 COLOR= C
60 FOR I = 0 TO 39
```

```
70 HLIN 0,39 AT I
80 NEXT I
90 END
```

60-80: riempimento dello schermo nel colore voluto.

Attenzione: è possibile anche scrivere:

```
70 VLIN 0,39 AT I
```

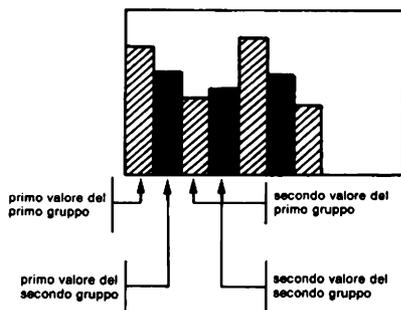
ESERCIZIO 4.6

Problema: stesso programma dell'Esercizio 4.4 ma per due gruppi di dieci valori. Si chiede cioè di fare un istogramma comparativo come sarebbe necessario, per esempio, per confrontare le vendite di dieci articoli in due periodi diversi.

Ingresso: due gruppi di dieci valori.

Uscita: l'istogramma comparativo verticale in bassa risoluzione (usare due colori diversi per le due sequenze di valori).

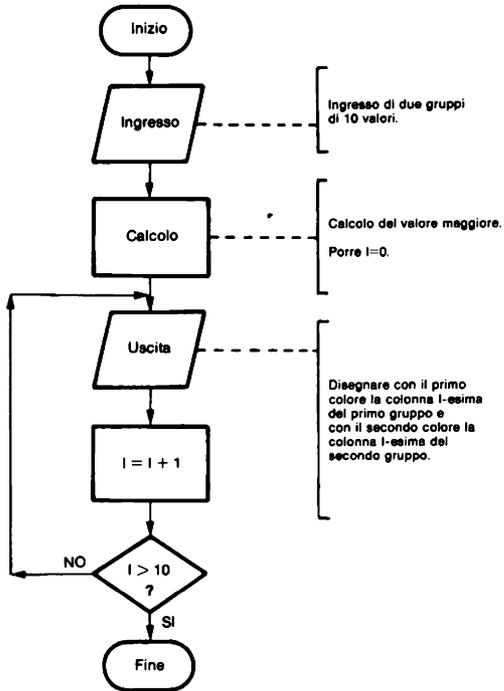
Esempio:



Analisi:

E' necessario, come nel caso precedente, fare corrispondere il valore massimo tra i due gruppi di valori alla massima ampiezza dell'istogramma.

Per registrare i valori si possono usare due tabelle, da dieci elementi, oppure una sola tabella con due indici.



ESERCIZIO 4.7

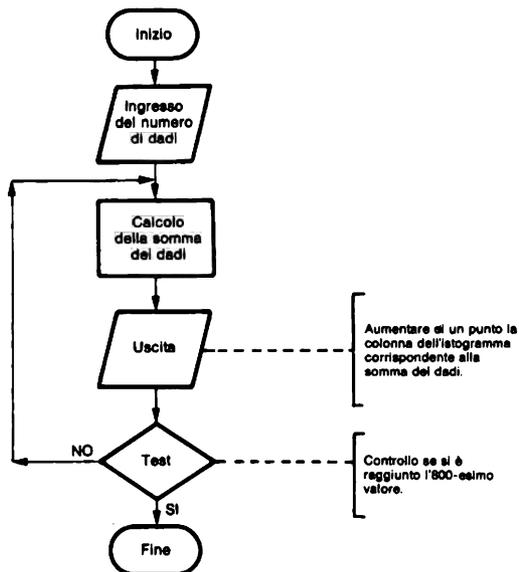
Problema: disegnare, in alta risoluzione, l'istogramma delle somme del lancio di n dadi per 800 lanci (vedere l'Esercizio 1.8).

Ingresso: il numero di dadi gettati assieme ogni volta.

Uscita: l'istogramma. Ciascuna colonna dell'istogramma corrisponde alle possibili somme per un numero fissato di dadi (per esempio con tre dadi si possono avere i valori somma: 3,4,5,...,17,18). Le ampiezze dell'istogramma dipendono dal numero di volte che le singole somme escono negli 800 lanci.

Analisi:

Riportiamo la parte principale del diagramma a blocchi.



ESEMPIO 4.8

Uscita: tracciare una linea a caso sullo schermo in alta risoluzione.

Soluzione:

```

10 HGR
20 HCOLOR= 3
30 X1 = RND (1) * 280
40 X2 = RND (1) * 280
50 Y1 = RND (1) * 160
60 Y2 = RND (1) * 160
65 :
70 HPLLOT X1, Y1 TO X2, Y2
80 END
  
```

Variabili:

X1,X2 - valore aleatorio delle ascisse

Y1,Y2 - valore aleatorio delle ordinate

Per tracciare la linea a caso vengono scelti due punti aleatori sullo schermo (linee 30 e 60) e poi vengono congiunti con una retta.

ESERCIZIO 4.9

Ingresso: coordinate di un punto dello schermo in alta risoluzione (due valori: il primo compreso tra 0 e 279 e l'altro tra 0 e 159).

Uscita: tracciare 100 linee a stella che colleghino questo punto con un altro scelto a caso.

E.SERCIZIO 4.10

Uscita: tracciare a caso delle linee sullo schermo, in alta risoluzione, in modo che il loro punto di mezzo sia sempre sul centro dello schermo (punto di coordinate 140,80).

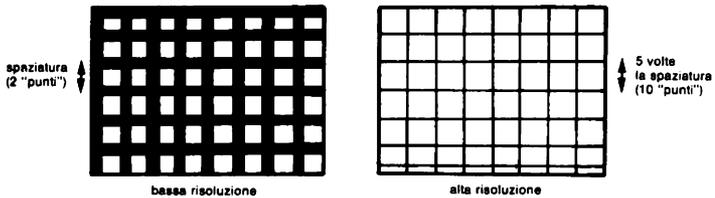
E.SERCIZIO 4.11

Problema: disegnare una quadrettatura dello schermo sia in bassa che in alta risoluzione.

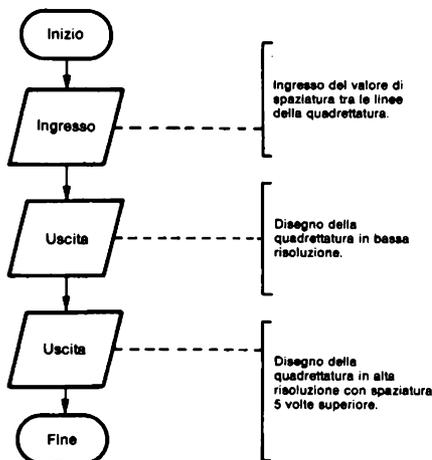
Ingresso: spaziatura (cioè numero di punti) entro ciascuna linea della quadrettatura.

Uscita: quadrettatura dello schermo grafico in bassa risoluzione con la spaziatura indicata in ingresso; successivamente stessa quadrettatura, ma in alta risoluzione con la spaziatura tra le linee cinque volte superiore.

*E*sempio: spaziatura di 2



Analisi:



La quadrettatura viene eseguita tracciando delle linee orizzontali e verticali separate dal valore di spaziatura dato.

ESERCIZIO 4.12

Problema: realizzare un gioco, con telecomando, in grafica a bassa risoluzione.

Ingresso: usare i due "paddle" del calcolatore.

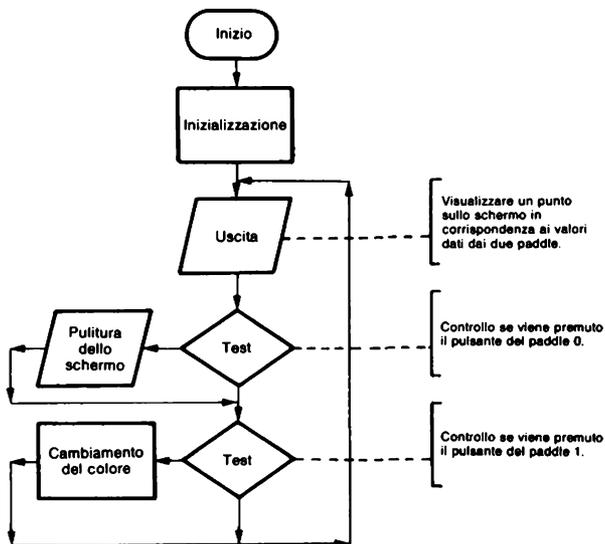
Uscita: visualizzare un punto le cui coordinate sono date tramite le manopole dei due "paddle". Cambiare il colore di visualizzazione se è premuto il pulsante del "paddle" 1; cancellate invece lo schermo se viene premuto il pulsante 0.

Analisi:

I valori dati dai due "paddle" vengono fatti corrispondere alle coordinate del punto.

I "paddle" generano un riferimento tra 0 e 255 che viene fatto corrispondere ad un valore tra 0 e 39 con la formula:

$$\text{PDL}(n) * 39 / 255$$



ESERCIZIO 4.13

Problema: stesso programma come il precedente 4.12, ma in alta risoluzione.

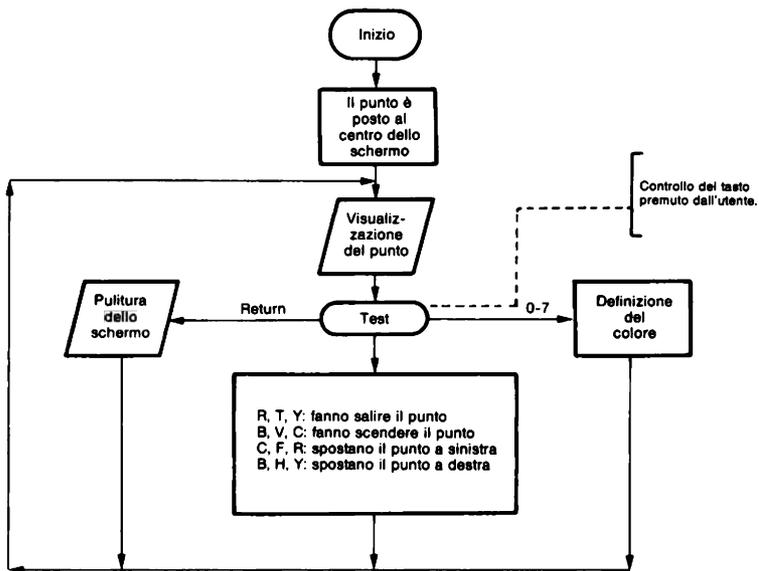
Ingresso: mediante la battitura di uno dei tasti R, T, Y, H, B, V, C e F si deve fare muovere il punto sullo schermo; una cifra tra 0 e 7 determina invece il colore e il tasto RETURN fa cancellare lo schermo.

Uscita: visualizzare il punto sullo schermo, con il colore indicato, e spostarlo in queste direzioni battendo la lettera corrispondente:

- R - in alto a sinistra
- T - in alto
- Y - in alto a destra
- H - a destra
- B - in basso a destra
- V - in basso
- C - in basso a sinistra
- F - a sinistra

Analisi:

Si deve stabilire la posizione iniziale sullo schermo; in questo caso l'abbiamo posta al centro.



ESEMPIO 4.14

Problema: tracciare una circonferenza con la grafica ad alta risoluzione.

Ingresso: le coordinate del centro della circonferenza e il suo raggio.

Uscita: tracciare sullo schermo la circonferenza voluta.

Analisi:

L'equazione della circonferenza di centro (0,0) e raggio R è:

$$Y^2 = R^2 - X^2$$

I punti della circonferenza possono allora essere ottenuti facendo variare X tra -R e R; a ciascun X corrisponderanno i due punti:

$$X, Y = \text{SQR}(R^2 - X^2) \text{ e } X, Y = -\text{SQR}(R^2 - X^2).$$

Nel caso il centro sia posto invece nel punto (CX,CY), i punti precedenti diventano:

$$CX + X, CY + \text{SQR}(R^2 - X^2) \text{ e } CX + X, CY - \text{SQR}(R^2 - X^2).$$

Soluzione:

```
10 TEXT HOME : VTAB 5
20 INPUT "COORDINATE CENTRO CIRC
   ONE. (X, Y): "; CX, CY
30 IF CX < 0 OR CX > 279 OR CY <
   0 OR CY > 159 THEN PRINT "X
   DA 0 A 279. Y DA 0 A 159": GOTO
   20
40 INPUT "RAGGIO: "; R: R = ABS (
   R)
45 IF CX - R < 0 OR CX + R > 279
   OR CY - R < 0 OR CY + R > 1
   59 THEN PRINT "IMPOSSIBILE"
   : GOTO 40
50 R2 = R ^ 2
60 HGR : HCOLOR= 3
70 HPLOT CX, CY
75 :
80 REM DISEGNO DELLA CIRCONFERE
   NZA
90 FOR X = 0 TO R
100 Y = SQR (R2 - X ^ 2)
110 HPLOT CX + X, CY + Y: HPLOT C
   X - X, CY + Y: HPLOT CX - X, C
   Y - Y: HPLOT CX + X, CY - Y
120 NEXT X
140 END
```

Variabili:

CX,CY - coordinate del centro

X - contatore

Y - variabile

45: se una parte della circonferenza esce dallo schermo, allora il programma viene ripetuto con un altro raggio.

50: calcolo del quadrato del raggio.

70: tracciamento di un punto nel centro della circonferenza.

90-130: tracciamento della circonferenza utilizzando la sua simmetria. Infatti se il centro è nell'origine e se un punto (X,Y) appartiene alla circonferenza, anche (-X,Y), (-X,-Y), (X,-Y), (Y,X), (-Y,X), (-Y,-X) e (Y,-X) appartengono alla stessa curva.

ESERCIZIO 4.15

Uscita: tracciare la funzione $Y=\text{SIN}(X)$ con la grafica ad alta risoluzione. Tracciare anche gli assi cartesiani prendendo come origine il punto dello schermo $X=140$ e $Y=80$.

Analisi:

Per tracciare una qualunque funzione sullo schermo è necessario scegliere una scala, cioè stabilire a quanti punti sullo schermo corrisponde una unità della funzione. Per esempio per la nostra funzione SIN abbiamo scelto che 30 punti sull'asse delle ordinate equivalgono al valore 1 del "seno".

Attenzione: poichè il centro degli assi è posto a (140,80), al punto (X,Y) dello schermo corrisponderà il punto (X-140,Y-80) della funzione da tracciare.



ESERCIZIO 4.16

Problema: tracciare una curva Y, funzione di X, definita dall'utente.

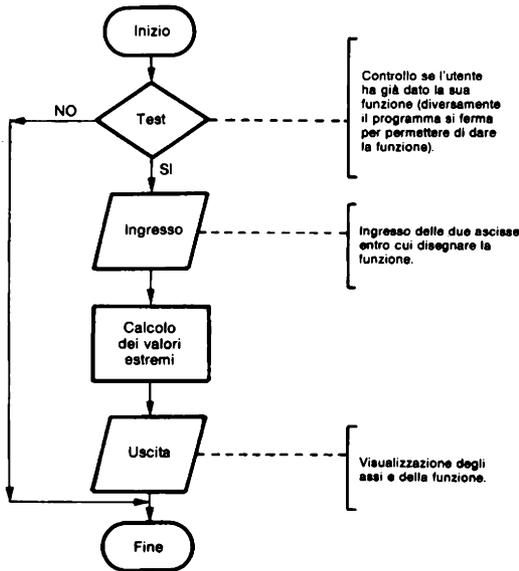
Analisi:

Come per l'Esercizio precedente (4.15) bisogna determinare la posizione dell'origine delle coordinate e la scala degli assi.

Si può anche chiedere all'utente quale porzione di ascisse desidera visualizzare e calcolare quindi sia l'ascissa dell'origine che la scala dell'asse delle ascisse.

Successivamente si devono calcolare i valori estremi della funzione, nell'intervallo desiderato, e determinare quindi sia il valore dell'ordinata dell'origine che la scala dell'asse delle ordinate.

Attenzione: l'utente deve fornire la funzione scelta sotto forma di una linea di programma.



ESERCIZIO 4.17

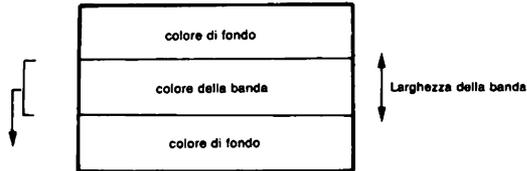
Uscita: una palla (rappresenta da un punto sullo schermo in bassa risoluzione) cade dall'alto dello schermo.

ESERCIZIO 4.18

Problema: stesso problema come il precedente Esercizio 4.17, ma per una banda orizzontale in bassa risoluzione.

Ingresso: colore di fondo, colore della banda e sua larghezza.

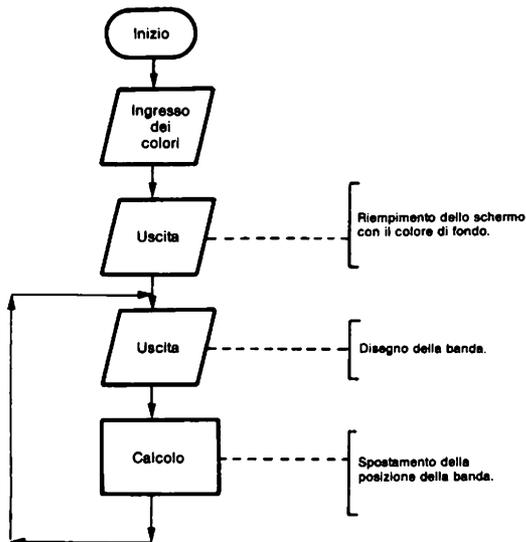
Uscita:



Analisi:

Per dare l'impressione che la banda si sposti verso il basso è sufficiente disegnare, sotto la banda, una nuova linea del suo stesso colore e subito dopo cancellare la prima linea della banda ricoprendola con una linea del colore del fondo. Questa operazione deve poi essere ripetuta più volte.

Quando la banda arriva sul fondo si tracciano le nuove linee a partire invece dall'alto.



ESERCIZIO 4.19

Problema: fare rimbalzare un rettangolo sui bordi dello schermo (grafica in alta risoluzione).

Ingresso: due valori numerici: H compreso tra 1 e 130 e L compreso tra 1 e 200.

Uscita: un rettangolo di altezza H e larghezza L animato da un movimento rettilineo in direzione aleatoria, rimbalza sui bordi dello schermo.

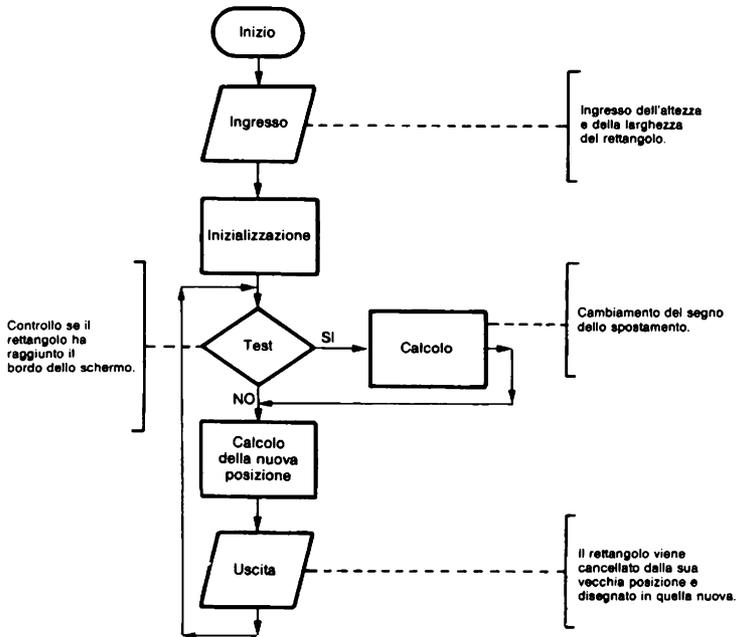
Analisi:

Per dare l'impressione che il rettangolo si muova si usa lo stesso metodo impiegato nell'Esercizio 4.18. Il rettangolo viene cancellato dalla sua vecchia posizione e poi disegnato nella nuova.

Per evitare che il movimento appaia a sbalzi, bisogna ridurre i tempi di calcolo e disegnare il nuovo rettangolo subito dopo aver cancellato il vecchio.

Per calcolare le coordinate successive di un punto, che si sposta in linea retta sullo schermo, può risultare comodo aggiungere alle sue coordinate due costanti contenenti gli spostamenti verticale e orizzontale.

Per farlo rimbalzare su uno dei bordi dello schermo, è sufficiente allora cambiare il segno dello spostamento corrispondente (spostamento orizzontale per i bordi destro e sinistro, spostamento verticale per i bordi superiore e inferiore).



GIOCHI GRAFICI, “FILETTO”, ESTRAZIONE DI UNA SERIE DI NUMERI ALEATORI, NUMERI PRIMI

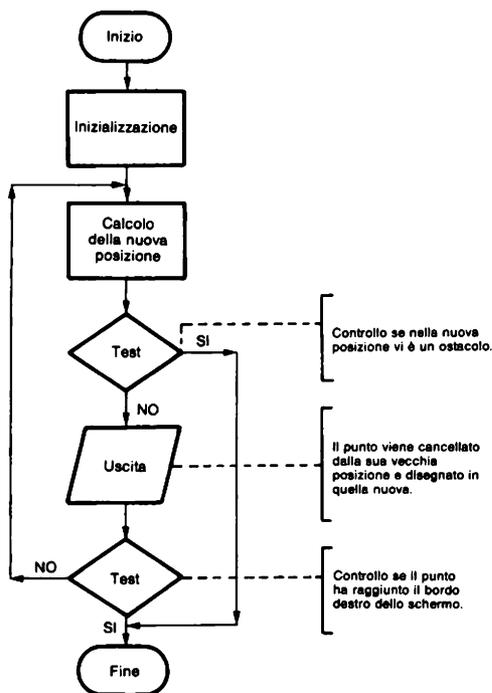
ESERCIZIO 5.1

- Problema:** è un gioco grafico in bassa risoluzione. Un punto sullo schermo si muove dal bordo sinistro a quello destro a velocità costante. Il gioco consiste nel fare arrivare il punto al bordo destro cercando di evitare degli ostacoli che sono posti a caso sullo schermo. Per questo motivo il giocatore può far deviare, verso l'alto o il basso, il punto usando la manopola di un “paddle”.
- Ingresso:** impiegare un “paddle” (opzionale: chiedere un livello iniziale di difficoltà del gioco).
- Uscita:** lo schermo grafico riempito di ostacoli mentre il punto si muove da sinistra verso destra. Se viene incontrato un ostacolo il gioco termina e poi si ripete.

Analisi:

Il programma legge il valore assunto dal “paddle”: se minore di 80 il punto viene fatto salire, se maggiore di 170 viene fatto invece scendere. (Attenzione a non uscire dallo schermo).

Opzione: il numero di punti ostacolo sullo schermo viene determinato in funzione di un certo livello di difficoltà.



ESERCIZIO 5.2

Problema: gioco del “filetto”. Su una scacchiera 3x3 ogni giocatore pone il suo segno in una casella. I due giocatori si alternano e vince chi riesce per primo a porre tre suoi segni in fila.

Ingresso: riga e colonna della casella scelta dal giocatore.

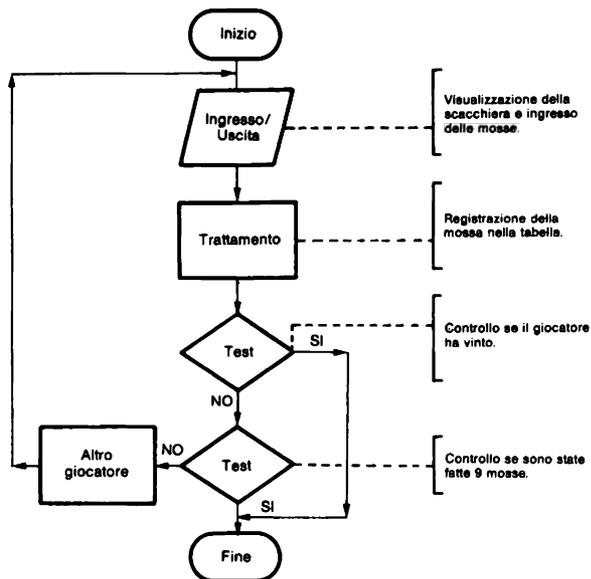
Uscita: visualizzazione della scacchiera con le mosse fatte, controllo dopo ogni mossa se uno dei due giocatori ha vinto e avviso finale se la partita è patta (dopo 9 mosse senza alcuna vincita).

Analisi:

La scacchiera è rappresentata da una tabella a tre linee e tre colonne.

Per indicare se una mossa corrisponde al primo o al secondo giocatore, è sufficiente porre, nella variabile che rappresenta la tabella, un 1 o un 2.

Dopo ogni mossa si deve controllare se su una riga o su una colonna o su una diagonale, della scacchiera, vi sono tre segni eguali.



ESERCIZIO 5.3

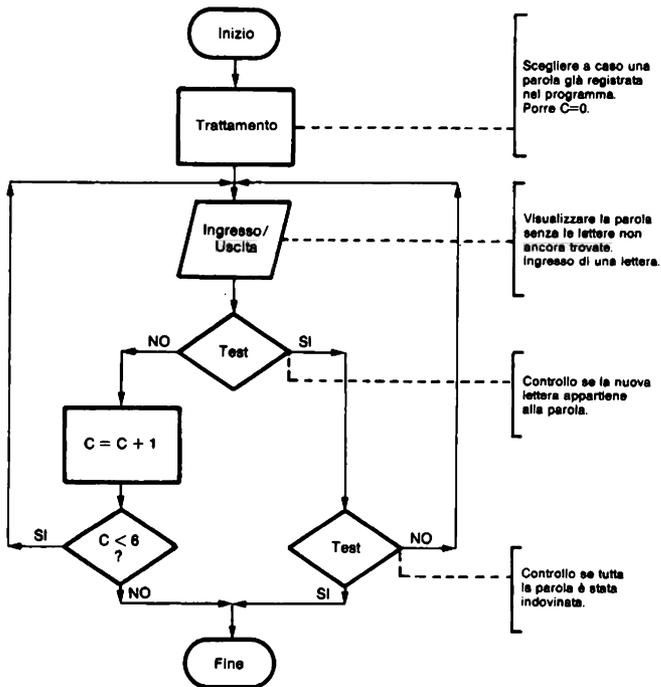
Problema: trovare una parola, di cui si conosce la lunghezza, proponendo di volta in volta le sue possibili lettere.

Ingresso: le lettere dell'alfabeto una alla volta per tentativi successivi.

Uscita: se la lettera proposta appartiene alla parola, essa viene visualizzata nella posizione corretta (o nelle posizioni se compare più di una volta). il giocatore non può fare più di cinque errori.

Analisi:

Registrare nel programma alcune parole da indovinare, mediante l'istruzione DATA.



ESERCIZIO 5.4

Problema: come l'Esercizio 1.7, ma all'inverso. Voi pensate un numero (tra 0 e 999999) e il calcolatore deve indovinarlo.

Ingresso: il giocatore deve rispondere ai tentativi del calcolatore con:

H - se il tentativo è troppo alto

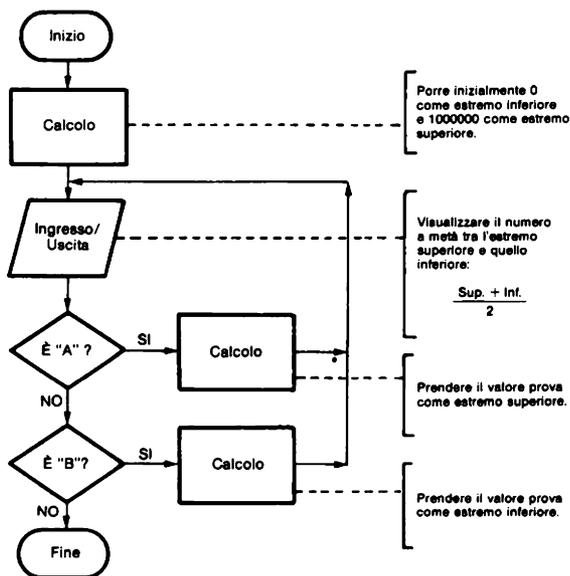
B - se è troppo basso

J - se il calcolatore ha trovato il numero esatto.

Uscita: i numeri tentativi.

Analisi:

Il metodo consiste nel proporre, come tentativo, il numero di mezzo dell'intervallo che si sa già contenere il valore esatto.



ESERCIZIO 5.5

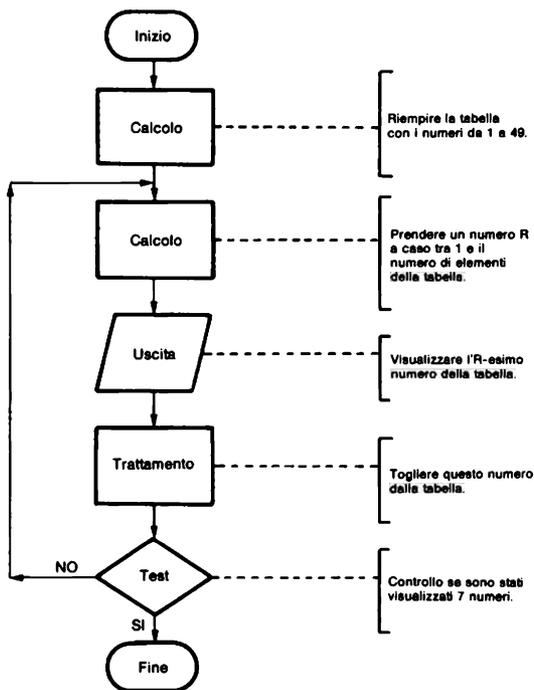
Problema: visualizzare una estrazione del gioco del lotto (versione francese).

Uscita: sette numeri interi aleatori compresi tra 1 e 49.

Analisi:

Per evitare che possa uscire due volte lo stesso numero, viene riempita una tabella con i numeri possibili (dall'1 al 49) e ogni volta che un numero "esce" esso viene tolto dalla tabella facendo rimontare i valori successivi (vedere l'Esercizio 2.8).

Attenzione: provate a riscrivere lo stesso programma per il lotto nella versione italiana: cinque numeri compresi tra 1 e 90.



ESERCIZIO 5.6

Problema: mescolare 52 carte.

Uscita: visualizzare le 52 carte mescolate.

Analisi:

E' sufficiente far corrispondere a ciascuna carta un numero compreso tra 0 e 51. Per mescolarle si può quindi usare il metodo dell'Esercizio precedente cioè estrarre le carte una alla volta a caso.

Per stabilire una corrispondenza tra i numeri e i segni delle carte si può porre:

da 0 a 12 : Picche

da 13 a 25 : Cuori

da 26 a 38 : Fiori

da 39 a 51 : Quadri

e nell'ambito di ciascun colore seguire l'ordine numerico delle carte. Per esempio:

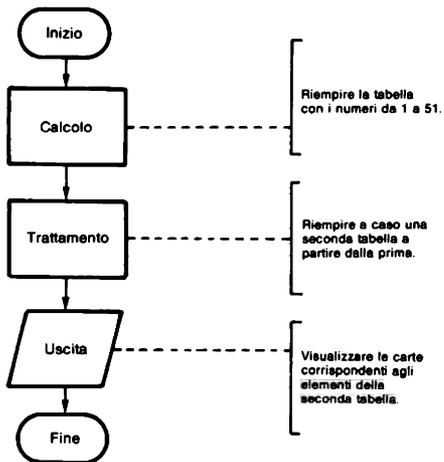
0 diviene l'Asso di Picche,

1 diviene il 2 di Picche,

13 diviene l'Asso di Cuori,

14 diviene il 2 di Cuori,

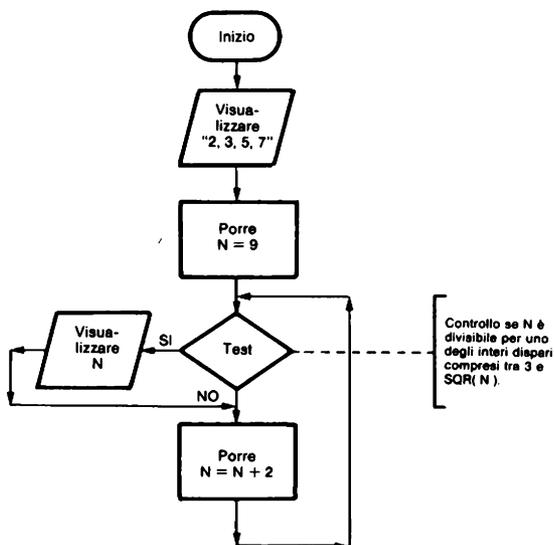
ecc.



ESERCIZIO 5.7

Uscita: visualizzare i numeri primi.

Analisi:



Un numero primo è un numero intero positivo divisibile solo per se stesso e per 1. Se vale l'eguaglianza $A = \text{INT}(A/B) \cdot B$ allora A è divisibile per B.

Per sapere se un numero è primo, è sufficiente fare il test precedente per tutti gli interi B compresi tra 1 e la radice quadrata del numero.

Attenzione: tutti i numeri pari eccetto il 2 non sono mai numeri primi (perchè sono ovviamente divisibili per 2) possono quindi essere eliminati dalla ricerca.

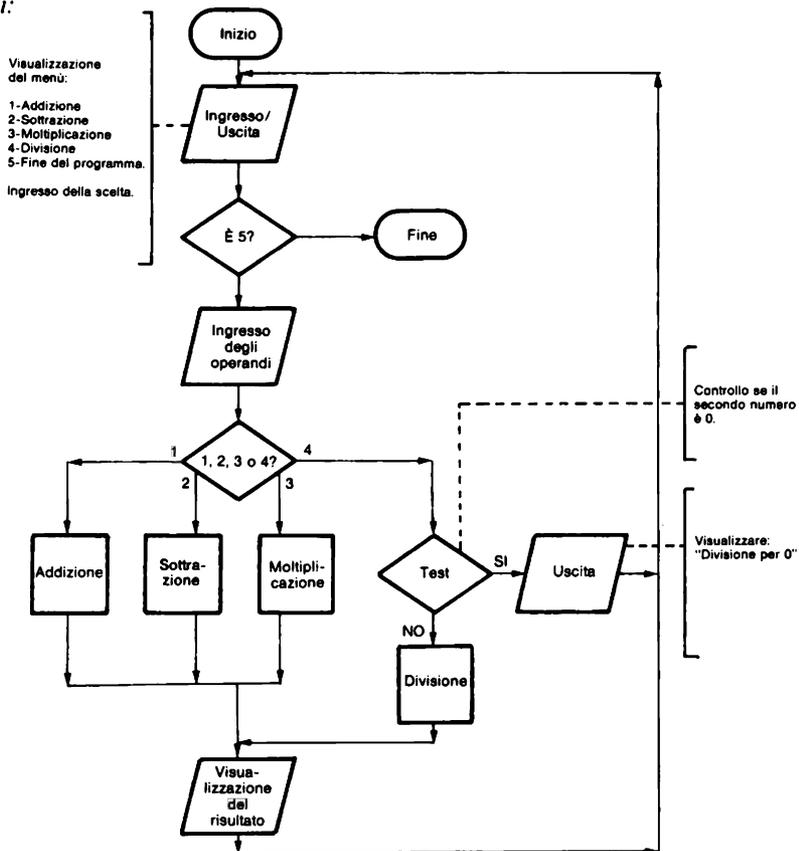
ESERCIZIO 5.8

Problema: eseguire una tra le operazioni di addizione, sottrazione, moltiplicazione e divisione su due numeri forniti dall'utente.

Ingresso: la scelta dell'operazione da fare e i due operandi.

Uscita: il risultato dell'operazione.

Analisi:



La visualizzazione del menù permette di scegliere tra più opzioni possibili. Il menù è costituito dalle singole opzioni precedute da una cifra o da una lettera; la scelta viene fatta battendo la cifra (o la lettera) scelta.

SCRITTURA E LETTURA DI DATI IN UN ARCHIVIO, PICCOLO PROGRAMMA DI GESTIONE DI INDIRIZZI

ESERCIZIO 6.1

Problema: registrare una lista di parole, battute alla tastiera, in un file sequenziale su dischetto.

Ingresso: il nome del file e una lista di parole.

Uscita: scrittura delle parole nel file.

Analisi:

Proponiamo due metodi per la scrittura del programma.

Con il primo metodo le parole sono scritte nel file una per una appena sono battute alla tastiera. Con il secondo invece esse sono dapprima registrate in una tabella, poi tutta la tabella viene trasferita nel file.

Il primo metodo presenta lo svantaggio di rallentare molto l'ingresso dei dati a causa dei tempi di registrazione di ogni parola sul dischetto. Il secondo invece è pericoloso perchè se dovesse mancare la corrente, dopo che avete caricato molti dati, dovrete rifare tutto daccapo.

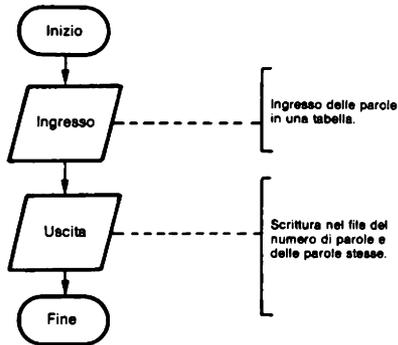
Il metodo migliore, per caricare un gran numero di dati, consiste ovviamente in una soluzione intermedia tra le due.

Attenzione: i due metodi sono analoghi a quelli utilizzati nell'Esercizio 2.4. Nel primo caso un "marker" (per noi la parola 'STOP') indica la fine dei dati; nel secondo invece viene espressamente indicato il numero di dati totale.

Primo metodo:



Secondo metodo:



ESERCIZIO 6.2

Problema: lettura della lista di parole registrata precedentemente.

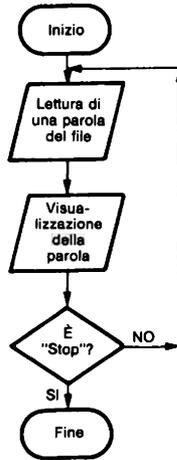
Ingresso: il nome del file creato nel programma 6.1.

Uscita: visualizzazione della lista di parole.

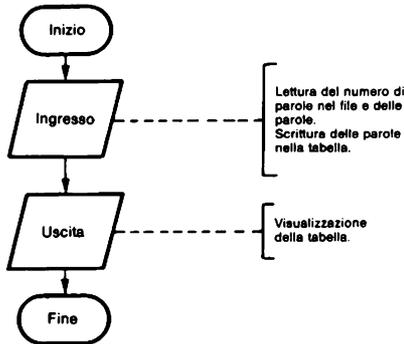
Analisi:

Proponiamo due metodi analoghi a quelli dell'esercizio precedente.

Primo metodo:



Secondo metodo:



ESERCIZIO 6.3

Problema: registrazione del nominativo e dell'indirizzo di una persona in un file ad accesso diretto.

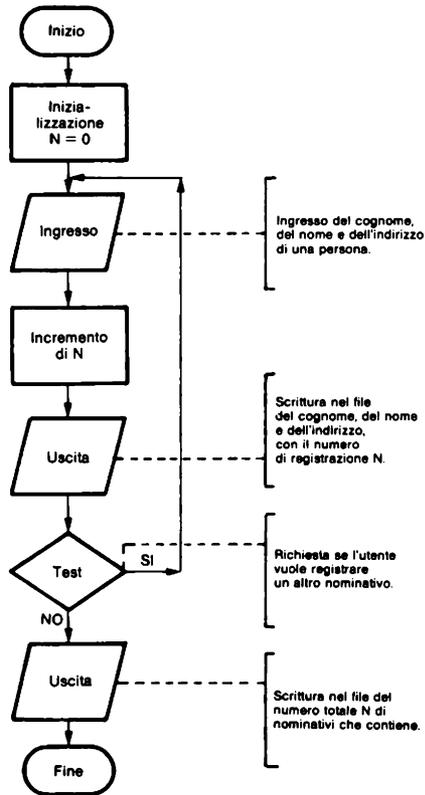
Ingresso: il cognome, il nome e l'indirizzo di più persone.

Uscita: registrazione dei dati nel file.

Analisi:

In un file ad accesso diretto è necessario precisare la lunghezza massima dei record, cioè del nominativo con il suo indirizzo più lungo. Attenzione però che bisogna riservare un carattere in più in ogni record per indicare la sua fine. Riteniamo che per registrare i nostri nominativi, 95 caratteri per ogni record siano sufficienti.

Attenzione: è utile registrare il numero di nominativi contenuti nel file.



ESERCIZIO 6.4

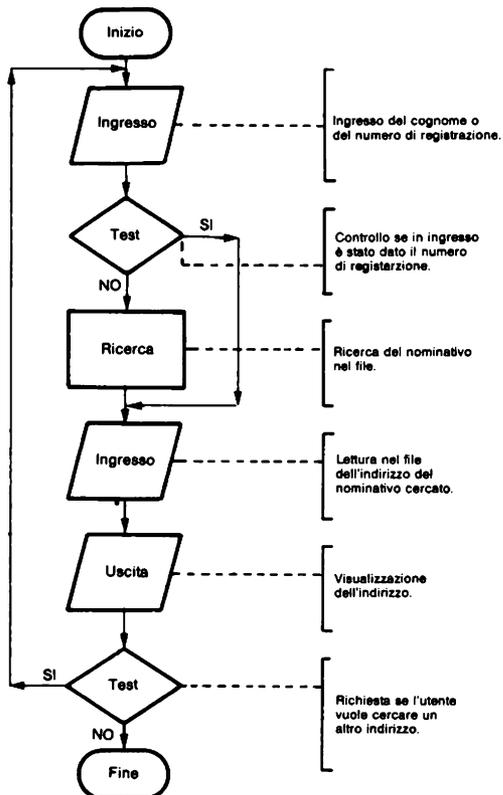
Problema: utilizzando il file creato in precedenza, ritrovare l'indirizzo di una persona a partire o dal suo numero di registrazione o dal suo cognome.

Ingresso: il cognome di una persona o il numero di registrazione.

Uscita: l'indirizzo di questa persona.

Analisi:

La ricerca per cognome viene fatta comparando un cognome alla volta. Attenzione che vi possono essere più persone con lo stesso cognome



SECONDA PARTE

PROGRAMMI

ESERCIZIO 1.2

```
10 HOME : VTAB 5
20 INPUT "QUALE TITOLO? "; T$
30 IF LEN (T$) > 40 THEN PRINT
   "TROPPO LUNGO!": GOTO 20
40 HOME : VTAB 10
50 HTAB 21 - LEN (T$) / 2
60 PRINT T$
70 VTAB 20: END
```

Commenti:

- 30: se il titolo è più lungo di 40 caratteri, ne viene richiesto un altro.
50: si prende la colonna di mezzo dello schermo (21) a cui si sottrae metà della lunghezza del titolo.

Esempio: titolo APPLE II

21	metà dello schermo
<u>-8/2</u>	metà della lunghezza
17	

- 70: VTAB 20 evita la riapparizione del cursore sulla linea successiva al titolo.

ESERCIZIO 1.3

```
10 HOME
20 INPUT "QUAL'E' IL VOSTRO NOME
   ? "; N$
30 INPUT "QUAL'E' IL VOSTRO ANNO
   DI NASCITA? "; A
40 PRINT
50 PRINT "BUON GIORNO "; N$
60 PRINT "NON DIRO' A NESSUNO CH
   E HAI "; 1982 - A;
70 PRINT " ANNI! CIAO"
80 END
```

Commenti:

Variabili:

N\$ - nome

A - anno di nascita

40: salta una linea per facilitare la lettura.

60: 1982 - A è l'età; (sostituire eventualmente 1982 con un nuovo anno).

ESERCIZIO 1.4

```
10 HOME
20 INPUT "DAI UN NUMERO? "; N
30 INPUT "QUALE PERCENTUALE? "; P

40 PRINT
50 DN = N * P / 100
60 PRINT "LA PERCENTUALE E' : "; D
   N
70 PRINT "IL VALORE MAGGIORATO E
   ' : "; N + DN
80 END
```

Commenti:

Variabili:

N - valore numerico

P - percentuale

DN - valore percentuale

50: calcolo del valore percentuale.

70: visualizzazione del valore maggiorato con la percentuale.

ESERCIZIO 1.5

```
10 HOME
20 VTAB 5
30 INPUT "QUAL'E' IL VOSTRO NOME
   ? "; N$
40 INPUT "QUANTO SPENDETE GIORNA
   LMENTE PER LA SPESA? "; D
```

```

50 DA = D * 365 * 1.07
55 :
60 HOME
70 VTAB 5
80 PRINT "HO CAPITO "; N$
90 VTAB 8
100 PRINT "ALLORA ANNUALMENTE SP
    ENDETE CIRCA "; DA; " LIRE !"
110 PRINT
120 PRINT
130 VTAB 15
140 PRINT "(LA MIA SPESA E' INVE
    CE ZERO LIRE...)"
150 END

```

Commenti:

Variabili:

D - spesa giornaliera

DA - spesa annuale

N\$ - nome

50: calcolo della spesa annuale.

ESERCIZIO 1.7

```

10 HOME : VTAB 5
20 REM INGRESSO DEL NUMERO DA I
    NDOVINARE
30 INPUT "QUAL'E' IL NUMERO DA I
    NDOVINARE (TRA 0 E 1000) ? "
    ; N
40 N = INT (N): IF N < 0 OR N >
    1000 THEN 30
45
50 REM INIZIO DEL GIOCO
60 HOME
70 PRINT
80 INPUT "PROVA: "; E
90 IF E > N THEN PRINT "TROPPO
    ALTO!": GOTO 70

```

```

100 IF E < N THEN PRINT "TROPPO
      BASSO!": GOTO 70
110 PRINT "BRAVO! HAI INDOVINATO
      "
120 END

```

Commenti:

Variabili:

N - numero da indovinare

E - valore prova

- 40: il numero da indovinare è un intero tra 0 e 1000. Dopo averlo reso sicuramente intero (con $N=INT(N)$), viene controllato se è compreso nell'intervallo 0 - 1000, diversamente viene richiesto un altro valore.
- 70: tra una prova e un'altra viene lasciata una linea vuota.
- 110: se i due test precedenti sono falsi, allora il numero è quello cercato per cui viene stampato un messaggio finale.

ESERCIZIO 1.8

```

10 HOME
20 INPUT "QUANTI DADI LANCIATE (
      DA 1 A 6)?":N
30 PRINT : IF N = 0 THEN END
40 IF N < 1 OR N > 6 THEN 20
50 S = 0
60 FOR I = 1 TO N
70 S = S + INT ( RND (1) * 6) +
      1
80 NEXT
90 PRINT "SOMMA = ";S
100 PRINT : PRINT : GOTO 20

```

Commenti:

Variabili:

I - contatore

N - numero di dadi lanciati assieme

S - somma dei dadi lanciati

- 30: se $N=0$ il programma si ferma; dopo una prova il giocatore può così proseguire o fermarsi.
- 40: questa linea deve essere posta dopo la 30 se no il programma non si ferma per $N=0$.

Attenzione: alla linea 50 la somma S viene posta inizialmente a 0 e poi, nel ciclo 60-80, viene calcolata per il lancio di N dadi.

ESERCIZIO 1.9

```

10 HOME
20 REM INIZIALIZZAZIONE DELLE
   VARIABILI
30 N = INT ( RND (1) * 1001)
40 C = 0
50 REM INIZIO DEL GIOCO
60 INPUT "PROVA: " : E
70 C = C + 1
80 IF E > N THEN PRINT "TROPPO
   ALTO": GOTO 50
90 IF E < N THEN PRINT "TROPPO
   BASSO": GOTO 50
100 PRINT : PRINT
110 PRINT "BRAVO! HAI INDOVINATO
   IN " : C : " PROVE"
120 END

```

Commenti:

Variabili:

C - contatore

N - numero da indovinare

E - prova

- 30: la variabile N assume un valore intero tra 0 e 1001. Infatti $RND(1)$ genera un numero a caso tra 0, compreso, e 1 non compreso. Moltiplicando per 1001 si ottiene un numero tra 0 e 1001 (non compreso) di cui viene presa solo la parte intera.
- 70: il contatore C viene "incrementato" di 1.
- 100: tra una prova e la successiva vengono lasciate due linee vuote.
- 110: C contiene il numero di prove.

ESERCIZIO 2.2

```
10 REM INGRESSO DEI DATI
20 HOME : VTAB 5
30 INPUT "PREZZO NETTO : ";P
40 PRINT
50 INPUT "CODICE IVA (A=7%, B=17
    6%, C=33%) : ";C$
60 REM RICERCA DEL VALORE IVA
70 IF C$ = "A" THEN C = 7: GOTO
    110
80 IF C$ = "B" THEN C = 17.6: GOTO
    110
90 IF C$ = "C" THEN C = 33: GOTO
    110
100 GOTO 40
105 :
110 REM CALCOLO E VISUALIZZAZIO
    NE
120 PRINT : PRINT
130 PT = P + P * C / 100
140 PT = INT (PT * 100) / 100
150 PRINT "PREZZO LORDO : ";PT
160 END
```

Commenti:

Variabili:

P - prezzo netto

PT - prezzo lordo con IVA

C - valore IVA

C\$ - codice IVA

50: i codici IVA sono riportati nella stringa dell'istruzione INPUT.

130: calcolo del prezzo lordo.

140: arrotondamento a due cifre decimali: il numero viene moltiplicato per 100 così che le due cifre che si vogliono conservare siano a sinistra del punto decimale (per esempio $6.7432 * 100$ diviene 674.32). Prendendo ora la parte intera di questo numero, si perdono le cifre che non interessano ($\text{INT}(674.32)=674$). Dividendo infine per 100 si ritrova il numero voluto arrotondato a due cifre (6.74).

Attenzione: il valore viene arrotondato "per difetto" con un errore tra 0 e 1 sull'ultima cifra decimale. Per ottenere un arrotondamento corretto è sufficiente scrivere:

140 PT=INT(P*100+.5)/100

ESERCIZIO 2.3

```
10 DIM P(10)
20 REM INGRESSO DEI DATI
30 HOME
40 FOR I = 1 TO 10
50 PRINT "PREZZO "; I; " ";
60 INPUT P(I)
70 NEXT I
80 PRINT
90 INPUT "CODICE IVA (A=7%, B=17
    %, C=33%) "; C$
95 :
100 REM RICERCA DEL VALORE IVA
105 :
110 IF C$ = "A" THEN C = 7: GOTO
    150
120 IF C$ = "B" THEN C = 17.6: GOTO
    150
130 IF C$ = "C" THEN C = 33: GOTO
    150
140 GOTO 80
145
150 REM CALCOLO
160 FOR I = 1 TO 10
170 P(I) = P(I) + P(I) * C / 100
180 P(I) = INT (P(I) * 100) / 10
    0
190 NEXT I
195 :
200 REM VISUALIZZAZIONE DEI PR
    EZZI

210 FOR I = 1 TO 10
220 PRINT "PREZZO "; I;
230 HTAB 30 - LEN ( STR$ ( INT
    (P(I)))): PRINT P(I)
240 NEXT I
250 END
```

Commenti:

Variabili:

I - contatore

C - valore IVA

C\$ - codice IVA

P(10) - tabella dei prezzi

10: dimensionamento della tabella.

40-70: ciclo d'ingresso dei prezzi. I dieci prezzi sono richiesti uno per uno alla linea 50. (L'istruzione INPUT "PREZZO",I,P(I) sarebbe gravemente errata perchè I verrebbe richiesto come dato d'ingresso).

160-190: calcolo dei dieci prezzi lordi che vanno a sostituire nella tabella i prezzi netti.

210-240: ciclo di visualizzazione.

230: l'allineamento sul punto decimale viene fatto visualizzando il prezzo a partire dalla colonna del punto, meno il numero di cifre davanti al punto stesso. Questo numero viene calcolato mediante la funzione LEN in questa maniera: LEN(STR\$(INT(PREZZO))).

Esempio: per 6454.54 LEN("6454") è 4

per 45.67 LEN("45") è 2

per cui la visualizzazione del primo numero inizia alla colonna $30-4=26$ e quella del secondo alla colonna $30-2=28$.

ESERCIZIO 2.4

Metodo 1

```
10 DIM L$(100)
20 READ N
30 FOR I = 1 TO N
40 READ L$(I)
50 NEXT I
60 FOR I = 1 TO N: PRINT L$(I): NEXT I

70 END
80 DATA 7
90 DATA MELA, TAVOLA, VETTURA
120 DATA ALBERO, CASA, SEDIA
130 DATA ALIANTE
```

Metodo 2

```
10 DIM L$(100)
20 FOR I = 1 TO 100
30 READ L$(I)
40 IF L$(I) < > "FINE" THEN NEXT
   I
50 N = I - 1
60 FOR I = 1 TO N: PRINT L$(I): NEXT

70 END
100 DATA MELA, TAVOLA, VETTURA
120 DATA ALBERO, CASA, SEDIA
130 DATA ALIANTE
500 DATA FINE
```

Commenti:

Variabili:

I - contatore

N - numero di parole contenuto nella lista

L\$(100) - tabella delle parole

Attenzione: i due metodi sono diversi nel modo come terminano la loro esecuzione. Il primo determina subito il numero di elementi nella lista con READ N a cui corrisponde DATA 7. Il secondo invece opera sino a che si trova la parola 'FINE' che noi abbiamo posto alla linea 500 con DATA FINE.

La linea 50 del secondo metodo è necessaria per determinare il numero di elementi della lista.

ESERCIZIO 2.5

```
10 DIM N$(50), P$(50)
20 REM INGRESSO DEI DATI
30 READ N
40 FOR I = 1 TO N
50 READ N$(I), P$(I)
60 NEXT I
70 HOME : VTAB 5
80 INPUT "COGNOME DA CERCARE :";
   N$
```

```

90 PRINT : INPUT "NOME "; P$
95 PRINT : PRINT
100 REM RICERCA
110 FOR I = 1 TO N
120 IF N$ = N$(I) AND P$ = P$(I)
    THEN PRINT P$: " "; N$: " E
    / NELLA TABELLA": GOTO 150
130 NEXT I
140 PRINT P$: " "; N$: " NON E' N
    ELLA TABELLA"
150 END
155 :
500 DATA 5
510 DATA ROSSI, CARLO, MERZARIO, G
    IOVANNI
520 DATA FRACCI, CARLA, PASSONI, L
    UIGI
530 DATA DORIA, ANDREA

```

Commenti:

Variabili:

I - contatore
N - numero delle persone
N\$ - cognome da cercare
P\$ - nome da cercare

N\$(50) - tabella dei cognomi

P\$(50) - tabella dei nomi

30-60: lettura dei cognomi e dei nomi con il primo metodo dell'Esercizio 2.4.
120: se viene trovato un cognome e un nome eguali, a quelli nella lista, viene visualizzato un messaggio di conferma e poi il programma salta alla linea 150 dove si ferma. Diversamente viene analizzato un altro nominativo (NEXT I).
140: se tutti i nominativi sono diversi, viene dato un messaggio di avviso.

Attenzione: N\$ e N\$(i) sono due variabili diverse. La prima è una variabile di stringa, la seconda rappresenta invece un elemento di una tabella di stringhe.

ESERCIZIO 2.6

```
10 DIM D(100)
20 HOME
30 INPUT "QUANTI NUMERI VUOI DAR
   E (MAX 100) ? ";N
40 PRINT
50 IF N < 1 OR N > 100 THEN 30
60 REM INGRESSO DEI DATI
70 FOR I = 1 TO N
80 PRINT "NUMERO ";I;
90 INPUT " ";D(I)
100 NEXT I
105 :
110 REM RICERCA DEL MIN E DEL M
    AX
120 MI = D(1):MA = D(1)
130 FOR I = 1 TO N
140 IF MI > D(I) THEN MI = D(I)
150 IF MA < D(I) THEN MA = D(I)
160 NEXT I
165 :
170 REM VISUALIZZAZIONE
180 HOME : VTAB 5
190 PRINT "MASSIMO : ";MA
200 PRINT "MINIMO : ";MI
210 END
```

Commenti:

Variabili:

I - contatore

N - numero di elementi nella tabella D(100)

MI - valore minore

MA - valore maggiore

D(100) - tabella di numeri

130-160: tutti gli elementi della tabella vengono confrontati con MI e con MA.
Se un elemento è più grande di MA, allora viene posto lui in MA. Se
invece un elemento è più piccolo di MI, viene allora posto in MI.

Attenzione: sarebbe possibile scrivere:

```
130 FOR I=2 TO N
```

perchè non è proprio necessario confrontare il primo elemento della tabella con MI e MA che già lo contengono. Ma in tal caso si avrebbe un errore per tabelle che hanno un solo elemento (N=1).

ESERCIZIO 2.7

```
10 DIM N$(1,50)
20 HOME
30 INPUT "QUANTE SONO LE PERSONE
   (DA 1 A 50) ? ";N
40 PRINT
50 IF N < 1 OR N > 50 THEN 30
55
60 REM INGRESSO DEI DATI
70 FOR I = 1 TO N
80 PRINT "PERSONA ";I; " "
90 INPUT " NOME      : ";N$(1,I)
100 INPUT " COGNOME  : ";N$(0,I)

110 PRINT : NEXT
115 :
120 REM RICERCA
130 M = 1
140 FOR I = 1 TO N
150 IF N$(0,I) = N$(0,M) AND N$(
   1,I) < N$(1,M) THEN M = I
160 IF N$(0,I) < N$(0,M) THEN M =
   1
170 NEXT I
175 :
180 REM VISUALIZZAZIONE
190 HOME : VTAB 5
200 PRINT "LA PRIMA PERSONA IN O
   RDINE ALFABETICO"
210 PRINT "E' QUELLA CON NUMERO
   ";M
220 PRINT : PRINT N$(1,M);" ";N
   $(0,M)
250 END
```

Commenti:

Variabili:

I - contatore

N - numero di persone

M - indice della prima persona in ordine alfabetico

N\$(1,50) - tabella dei cognomi (N\$(0,x)) e dei nomi (N\$(1,x))

80: viene visualizzato il numero della persona in ingresso.

Attenzione: viene utilizzata una tabella a due indici; il primo indice è 0 per i cognomi e 1 per i nomi mentre il secondo indice indica i vari nominativi.

	0	1
1	nome 1	cognome 1
2	nome 2	cognome 2
3	-	-
4	-	-
5	-	-
6	-	-
N\$ (1,50)		

E' possibile unire le istruzioni 150 e 160 in una sola:

```
150 IF N$(0,I)<N$(0,M) OR ( N$(0,I)=N$(0,M) AND N$(1,I)<N$(1,M) ) THEN  
M=I
```

ESERCIZIO 2.8

Metodo 1

```
10 DIM TA(50), TB(50)  
20 REM INGRESSO DEI DATI  
30 READ N1  
40 FOR I = 1 TO N1: READ TA(I): NEXT  
I  
45 :  
50 REM ORDINAMENTO  
60 N2 = 0
```

```

70  REM  RICERCA DELL'ELEMENTO PI
    U' GRANDE
75  REM  DELLA PRIMA TABELLA
80  N2 = N2 + 1
90  M = 1
100 FOR I = 1 TO N1
110  IF TA(I) > TA(M) THEN M = I
120  NEXT
125 :
130  REM  INSERIMENTO NELLA SECON
    DA TABELLA
140  TB(N2) = TA(M)
150  TA(M) = - 1
160  IF N2 < N1 THEN 80
165 :
170  REM  VISUALIZZAZIONE
180  FOR I = 1 TO N2: PRINT TB(I)
    : NEXT I
190  END
195 :
500  DATA  12
510  DATA  3, 10, 15, 1000, 5, 477, 345
    , 37, 50, 6, 74, 67

```

Metodo 2

```

10  DIM TA(50), TB(50)
20  REM  INGRESSO DEI DATI
30  READ N1
40  FOR I = 1 TO N1: READ TA(I): NEXT
    I
45 :
50  REM  ORDINAMENTO
60  N2 = 0
65 :
70  REM  RICERCA DELL'ELEMENTO PI
    U' GRANDE
75  REM  DELLA PRIMA TABELLA
80  N2 = N2 + 1
90  M = 1
100 FOR I = 1 TO N1

```

```

110 IF TA(I) > TA(M) THEN M = I
120 NEXT
125 :
130 REM SCAMBIO DEGLI ELEMENTI
140 TB(N2) = TA(M)
150 IF M < N1 THEN FOR I = M +
      1 TO N1:TA(I - 1) = TA(I): NEXT
      I
160 N1 = N1 - 1
170 IF N1 > 0 THEN GO
175 :
180 REM VISUALIZZAZIONE
190 FOR I = 1 TO N2: PRINT TB(I)
      : NEXT I
200 END
205 :
500 DATA 12
510 DATA 3, 10, 15, 1000, 5, 477, 345
      , 37, 50, 6, 74, 67

```

Commenti:

Variabili:

I - contatore

N1 - numero di elementi della prima tabella

N2 - numero di elementi della seconda tabella

M - indice dell'elemento più grande della prima tabella

TA(50) - prima tabella: numeri da ordinare

TB(50) - seconda tabella: numeri ordinati

Attenzione: i dati sono letti con il primo metodo dell'Esercizio 2.4.

L'istruzione DATA della linea 500 indica il numero degli elementi da leggere, mentre la DATA della linea 510 contiene i numeri da ordinare.

ESERCIZIO 2.9:

```

10 DIM N$(1, 50)
20 REM INGRESSO DEI DATI
30 HOME : INPUT "QUANTE PERSONE
      ? "; N

```

```

35 PRINT
40 IF N < 0 OR N > 50 THEN 30
50 FOR I = 1 TO N
60 PRINT "PERSONA "; I
70 INPUT "COGNOME : "; N$(0, I)
80 INPUT "NOME : "; N$(1, I)
90 PRINT : NEXT I
95 :
100 REM ORDINAMENTO
110 P = 1
120 M = P
130 FOR I = P TO N
140 IF N$(0, I) < N$(0, M) OR (N$(
    0, I) = N$(0, M) AND N$(1, I) <
    N$(1, M)) THEN M = I
150 NEXT I
160 N$ = N$(0, P):P$ = N$(1, P)
170 N$(0, P) = N$(0, M):N$(1, P) = N
    $(1, M)
180 N$(0, M) = N$:N$(1, M) = P$
190 P = P + 1: IF P < N THEN 120
195 :
200 REM VISUALIZZAZIONE
210 FOR I = 1 TO N: PRINT N$(1, I
    ), N$(0, I): NEXT I
220 END

```

Commenti:

Variabili:

I - contatore

N - numero delle persone

P - indice a partire dal quale i nominativi sono ordinati

M - indice della prima persona in ordine alfabetico

N\$,P\$ - variabili

N\$(1,50) - tabella dei cognomi e dei nomi (N\$(0,x) sono i cognomi e N\$(1,x) sono i nomi).

120-150: ricerca della prima persona in ordine alfabetico successiva alla P-esima.

160-180: scambio della P-esima persona con la M-esima.

ESERCIZIO 2.10

```
10 DIM N$(1, 50)
20 READ N
30 FOR I = 1 TO N: READ N$(0, I),
   N$(1, I): NEXT I
40 :
50 REM INGRESSO DEL NOMINATIVO
   DA AGGIUNGERE
60 HOME : VTAB 5
70 INPUT "NOMINATIVO DA AGGIUNGE
   RE (BATTERE 'FINE' PER TERMI
   NARE): "; N$
80 IF N$ = "FINE" OR N = 50 THEN
   END
90 PRINT : INPUT "NOME ? "; P$
95 :
100 REM RICERCA
110 FOR I = 1 TO N
120 IF N$ < N$(0, I) OR (N$ = N$(
   0, I) AND P$ < N$(1, I)) THEN
   180
130 NEXT
135 :
140 REM AGGIUNTA ALLA FINE DELL
   A TABELLA
150 N$(0, N + 1) = N$: N$(1, N + 1) =
   P$
160 GOTO 230
170 :
180 REM INSERIMENTO
190 FOR J = N TO I STEP - 1
200 N$(0, J + 1) = N$(0, J): N$(1, J +
   1) = N$(1, J)
210 NEXT J
220 N$(0, I) = N$: N$(1, I) = P$
230 N = N + 1
240 HOME : VTAB 5
250 FOR I = 1 TO N: PRINT N$(0, I
   ), N$(1, I): NEXT I
```

```

260 PRINT : PRINT : GOTO 70
265 :
500 DATA 5
510 DATA AMALDI, UGO, FERMI, ENRIC
    O, MAJORANA, ETTORE
520 DATA PEANO, GIUSEPPE, RIGHI, A
    UGUSTO

```

Commenti:

Variabili:

I, J - contatori

N - numero delle persone

N\$, P\$ - cognome e nome della persona da inserire

N\$(1,50) - tabella dei cognomi e dei nomi (N\$(0,x) sono i cognomi e N\$(1,x) sono i nomi).

70 e 260: l'utente può inserire più nominativi; per terminare deve battere 'FINE'.

80: la tabella non può contenere più di 50 nominativi.

110-130: se la posizione d'inserimento non è l'ultima, il programma prosegue alla linea 180 e la variabile I contiene tale posizione. Nel caso invece che la posizione sia l'ultima il ciclo termina ed il programma prosegue alla linea 140.

230: N viene incrementato perchè la tabella contiene una persona in più.

ESERCIZIO 3.1

```

10 HOME : VTAB 5
20 INPUT "BATTETE UNA LINEA : ";
    A$
30 IF RIGHT$(A$,1) = " " THEN
    A$ = MID$(A$,1, LEN(A$) -
    1): GOTO 30
40 PRINT "("; A$; ")"
50 END

```

Commenti:

30: se l'ultimo carattere di A\$ è uno spazio, allora esso viene soppresso prendendo tutti i caratteri di A\$ salvo l'ultimo (MID\$(A\$,1,LEN(A\$)-1)); la prova viene poi ripetuta sino a che si hanno spazi in coda alla linea.

Esempio: se A\$="LIBRO" si ottiene:

A\$=MID\$(A\$,1,5)="LIBRO"

40: A\$ viene visualizzata tra parentesi per confermare che gli spazi in coda sono stati tolti.

Attenzione: quando viene eseguita una istruzione INPUT, gli spazi dati all'inizio della risposta non sono presi in considerazione. Per esempio se battete "GIORNO", alla variabile corrispondente viene attribuito il valore "GIORNO".

È possibile scrivere:

```
30 IF RIGHT$(A$,1)=" " THEN A$ = LEFT$(A$,LEN(A$)-1) : GOTO 30
```

ma nel caso A\$ sia una stringa vuota si commetterebbe un errore. E' infatti possibile scrivere MID\$(A\$,1,0), che da la stringa vuota, ma non è invece possibile scrivere LEFT\$(A\$,0).

ESERCIZIO 3.2

```
10 HOME : VTAB 5
20 INPUT "BATTETE UNA LINEA : ";
   A$
30 I = 1
40 IF MID$(A$,I,1) = " " THEN
   A$ = MID$(A$,1,I - 1) + MID$(
   A$,I + 1, LEN (A$)): GOTO 4
   0
50 I = I + 1: IF I < = LEN (A$)
   THEN 40
60 PRINT "("; A$; ")"
70 END
```

Commenti: Tutti i caratteri della stringa sono confrontati uno per uno con lo spazio (linea 40). Se l'I-esimo carattere è uno spazio, esso viene soppresso e il programma

rimane sulla stessa linea 40 poichè l'I-esimo carattere è ora cambiato. Diversamente I viene incrementato. Il processo continua sino a che I diviene maggiore della lunghezza della stringa.

Attenzione: sconsigliamo di utilizzare il ciclo:

```
30 FOR I=1 TO LEN(A$)
40 ...
50 NEXT I
```

perchè LEN(A\$) sarebbe calcolata solo per il valore iniziale della stringa.

ESERCIZIO 3.3

```
10 HOME : VTAB 5
20 INPUT "BATTETE UNA PAROLA "; A
   $
30 IF LEN (A$) < 2 THEN 70
40 FOR I = LEN (A$) - 1 TO 1 STEP
   - 1
50 A$ = MID$ (A$, 1, I) + " " + MID$
   (A$, I + 1)
60 NEXT I
70 PRINT A$
80 END
```

Commenti:

30: è impossibile inserire degli spazi se la lunghezza della parola è 0 o 1.

Attenzione: gli spazi sono aggiunti a partire dalla fine della stringa perchè la sua lunghezza va aumentando. Sarebbe possibile scrivere:

```
40 FOR I=1 TO LEN(A$)*2-2 STEP 2
50 ...
60 NEXT I
```

ESERCIZIO 3.4

```
10 HOME : VTAB 5
20 INPUT "BATTETE UNA FRASE "; P$
```

```

30 PRINT
40 PRINT P$
50 P$ = MID$(P$,2)
60 IF LEN (P$) > 0 THEN 40
70 END

```

Commenti:

50: MID\$(P\$2) restituisce tutti i caratteri di P\$ salvo il primo.

ESERCIZIO 3.5

```

10 HOME : VTAB 5
20 INPUT "BATTETE UNA FRASE :";P
   $
30 REM INIZIO DEL CICLO
40 PRINT P$
50 FOR I = 1 TO LEN (P$)
60 IF MID$(P$,I,1) = " " THEN
   P$ = MID$(P$,I + 1): GOTO
   40
70 NEXT I
80 END

```

Commenti:

Variabili:

I - posizione del primo spazio di P\$ incontrato da sinistra

P\$ - variabile

ESERCIZIO 3.7

Metodo 1

```

10 HOME : VTAB 5
20 INPUT "LINEA (40 CAR. MAX) :
   ";L$
30 IF LEN (L$) > 40 THEN 20
35 :
40 REM CREAZIONE DELLA LINEA DI
   40 CARATTERI

```

```

50 S$ = ".....
      .....
60 L$ = L$ + MID$(S$,1,40 - LEN
      (L$))
65 :
70 REM SCORRIMENTO
80 VTAB 15: PRINT L$
90 L$ = RIGHT$(L$,39) + LEFT$
      (L$,1)
100 FOR I = 1 TO 80: NEXT I
110 GOTO 80

```

Commenti:

- 50-60: la linea viene completata con dei punti perchè contenga 40 caratteri.
 90: il primo carattere della linea viene messo nell'ultima posizione.
 100: ciclo d'attesa.

Metodo 2

```

10 HOME : VTAB 5
20 INPUT "LINEA (40 CAR. MAX) :
      ":L$
30 IF LEN(L$) > 40 THEN 20
35 :
40 REM SCORRIMENTO
50 FOR I = 40 TO 1 STEP - 1
60 VTAB 15: PRINT MID$(L$,41 -
      I): TAB(I): MID$(L$,1,40 -
      I): SPC(40)
65 :
70 PRINT
80 FOR J = 1 TO 80: NEXT J
90 NEXT I
100 GOTO 50

```

Commenti:

- 60: visualizzazione, a partire dalla prima colonna dello schermo, della parte della linea che ha superato, il bordo destro, poi dalla colonna I in avanti l'inizio della linea sino al bordo destro.

ESERCIZIO 3.8

```
10 HOME : VTAB 5
20 PRINT "BATTETE UNA FRASE NEGA
   TIVA : "
30 PRINT : INPUT A$
40 REM INIZIO DEL CICLO
45 REM DI CONTROLLO
50 I = 1
60 IF MID$ (A$, I, 5) = " NON " THEN
   A$ = MID$ (A$, 1, I) + MID$
   (A$, I + 5)
90 REM TEST DI FINE DEL CICLO
95 REM DI CONTROLLO
100 I = I + 1: IF I < LEN (A$) -
   4 THEN 60
110 IF LEFT$ (A$, 4) = "NON " THEN
   A$ = RIGHT$ (A$, LEN (A$) -
   4)
120 PRINT : PRINT "LA FRASE IN P
   OSITIVO E' : "
130 PRINT A$
150 END
```

Commenti:

Variabili:

I - contatore

A\$ - variabile

60: viene ricercata la parola 'NON' per essere soppressa e rendere il senso della frase affermativo.

110: si ricerca anche NON all'inizio della frase.

Attenzione: non abbiamo utilizzato un ciclo FOR-NEXT per la stessa ragione indicata nell'Esercizio 3.2.

MID\$(A\$,I,I) da i primi caratteri di A\$ sino all'I-esimo incluso.

ESERCIZIO 3.9

```
10 HOME : VTAB 5
20 INPUT "DAI UNA STRINGA DI CAR
   ATTERI : "; C$
30 PRINT : PRINT
40 INPUT "E I CARATTERI DA INSER
   IRE : "; I$
50 INPUT "NELLA POSIZIONE : "; P
60 IF P < 0 OR P > LEN (C$) THEN
   50
70 C$ = MID$ (C$,1,P) + I$ + MID$
   (C$,P + 1)
80 PRINT : PRINT
90 PRINT "LA STRINGA COMPLETA E'
   : "
100 PRINT C$
200 END
```

Commenti:

60: controlla la validità della posizione d'inserimento data in ingresso.
70: inserimento.

Esempio: da "APPLE II" con I\$="PLUS" e P=6,
si ottiene:

MID\$(C\$,1,P)="APPLE" e
MID\$(C\$,P,1)="II"

ESERCIZIO 3.10

```
10 HOME VTAB 5
20 INPUT "BATTETE UNA LINEA DI T
   ESTO (80 CAR MAX) : ".L$
30 IF L$ = "" OR LEN (L$) > 80 THEN
   20
40 PRINT : PRINT "BATTETE LA PAR
   OLA DA TOGLIERE E QUELLA DA
   METTERE."
45 PRINT
50 INPUT "PAROLA DA TOGLIERE : "
   : R$
```

```

55 INPUT "PAROLA DA METTERE :".M
   $
60 IF LEN (R$) > 80 OR LEN (M$
   ) > 80 THEN 40
65
70 REM INIZIO DELLA RICERCA
80 I = 1:F = 0:R = LEN (R$):M =
   LEN (M$)
90
100 IF MID$ (L$,I,R) < > R$ THEN
   210
110 F = 1
115 PRINT
120 IF LEN (L$) - R + M > 80 THEN
   PRINT "SOSTITUZIONE IMPOSSI
   BILE!": GOTO 300
125
130 REM CONFERMA
140 PRINT MID$ (L$,1,I - 1):
150 INVERSE PRINT R$: NORMAL

160 PRINT MID$ (L$,I + R)
170 PRINT PRINT "CONFERMATE LA
   SOSTITUZIONE (S/N)? ".
180 GET A$: PRINT A$: IF A$ < >
   "S" AND A$ < > "N" THEN PRINT
   CHR$ (7): GOTO 180
190 IF A$ = "N" THEN 210
200 L$ = MID$ (L$,1,I - 1) + M$ +
   MID$ (L$,I + R)
210 I = I + 1: IF I < = LEN (L$
   ) - R + 1 THEN 90
215 :
300 REM VISUALIZZAZIONE
310 HOME VTAB 5
320 IF F = 0 THEN PRINT "LA PAR
   OLA NON E' NELLA LINEA": PRINT

330 PRINT L$
350 END

```

Commenti:

Variabili:

I - contatore

F - "flag" (0 se la parola da togliere non è nella linea)

R\$ - lunghezza di R\$

M\$ - lunghezza di M\$

L\$ - linea di testo

R\$ - parola da togliere

M\$ - parola da mettere

- 30: se la linea non contiene alcuna parola, la sua elaborazione non è possibile.
- 80: inizializzazione del contatore, del flag e della lunghezza delle parole.
- 100: controllo se la parola da togliere si trova nella posizione I; se negativo allora il contatore I viene incrementato alla linea 210.
- 110: se positivo, il flag viene posto eguale a 1 per indicare che la parola esiste.
- 140-160: visualizzazione della linea con la parola da togliere in inverso. Se la parola è presente più volte essa viene sostituita in più fasi successive.
- 180: se l'utente non risponde con S o N, allora il calcolatore emette un suono (PRINT CHR\$(7)).
- 320: se il flag F non è rimasto eguale a 0, vuol dire che la parola è stata trovata ed è stata eseguita la linea 110.

ESERCIZIO 3.11

```
10 HOME
20 INPUT "DATE UNA LINEA DI TEST
   0                ";L$
30 IF L$ = "" THEN 20
25 :
40 REM CREAZIONE DELLA STRINGA
   INVERSA
50 I$ = ""
60 FOR I = LEN (L$) TO 1 STEP -
   1
70 I$ = I$ + MID$ (L$, I, 1): NEXT
   I
75 :
80 REM CREAZIONE DELLA STRINGA
   ALEATORIA
```

```

90 T$ = L$:A$ = ""
100 FOR I = 1 TO LEN (L$)
110 A = INT ( RND (1) * LEN (T$
    ) ) + 1
120 A$ = A$ + MID$ (T$,A,1)
130 T$ = MID$ (T$,1,A - 1) + MID$
    (T$,A + 1)
140 NEXT I
145 :
150 REM  VISUALIZZAZIONE
160 HOME : VTAB 5: PRINT L$
170 PRINT : PRINT I$
180 PRINT : PRINT A$
200 END

```

Commenti:

Variabili:

I - contatore

A - posizione aleatoria di una lettera in T\$

L\$ - linea di testo

I\$ - linea inversa

A\$ - linea aleatoria

T\$ - variabile

90: la linea di testo viene ricopiata in T\$ per potere essere elaborata senza perdere l'originale. A\$ viene inizializzata.

110: INT(RND(1)*LEN(T\$))+1 da un numero aleatorio tra 1 e la lunghezza di T\$.

120-130: il carattere A-esimo di T\$ viene aggiunto ad A\$ poi è soppresso da T\$.

Attenzione: è possibile scrivere:

```

60 FOR I=1 TO LEN(L$)
70 I$=MID$(L$,I,1) + I$ : NEXT I

```

ESERCIZIO 3.13

```
10 HOME YTAB 5
20 PRINT "BATTETE UNA FRASE (POT
    ETE USARE ANCHE LE VIRGOLE E
    I DUE PUNTI) "
30 P$ = "" : I = 0
40 REM INGRESSO DELLA FRASE
50 GET A$: PRINT A$.
60 IF ASC (A$) < > 8 THEN 100
70 REM "FRECCIA A SINISTRA"
80 IF I > 0 THEN I = I - 1 : P$ =
    MID$ (P$, 1, I)
90 PRINT " ": CHR$ (8) : GOTO 50

100 IF ASC (A$) < > 13 THEN P$
    = P$ + A$ : I = I + 1 : GOTO 5
    0
110 REM "RETURN"
120 PRINT PRINT
130 PRINT "LA FRASE E' "
140 PRINT P$
150 END
```

Commenti:

Variabili:

I - numero di caratteri di P\$

P\$ - linea battuta alla tastiera

A\$ - un tasto battuto alla tastiera

30: inizializzazione.

80: non è possibile togliere un carattere a P\$ se non ne contiene alcuno (I=0).

90: prima di saltare alla 50, l'ultimo carattere viene cancellato dallo schermo.

100: il carattere viene aggiunto alla fine di P\$ (se diverso da RETURN).

ESERCIZIO 3.14

```
10 DIM T(50)
15 HOME : VTAB 3
20 REM INGRESSO DEI DATI
30 INPUT "QUANTI VALORI DATE (DA
    1 A 50)? ";N
40 IF N < 1 OR N > 50 THEN 30
50 FOR I = 1 TO N: PRINT "VALORE
    ";I;
60 INPUT T(I): NEXT I
65 REM CALCOLO DELLE PERCENTUAL
    I
70 S = 0
80 FOR I = 1 TO N:S = S + T(I): NEXT
    I
90 FOR I = 1 TO N:T(I) = T(I) /
    S * 100: NEXT I
100 HOME : VTAB 5
110 FOR I = 1 TO N
120 PRINT "VALORE ";I" : "; INT
    (T(I) * 100 + .5) / 100; " %"

130 NEXT I
150 END
```

Commenti:

Variabili:

I - contatore

N - numero di valori

S - somma dei valori

T(50) - tabella dei valori

Per calcolare le percentuali, si calcola dapprima la somma di tutti i valori (linea 80), poi si divide ogni valore per questa somma e si moltiplica per 100 (linea 90). I risultati sono arrotondati a due cifre decimali (linea 120).

ESERCIZIO 3.15

```
10 DIM L(26), T$(30)
20 REM INGRESSO DEL TESTO
30 HOME : VTAB 5
40 INPUT "NUMERO DELLE LINEE (DA
   1 A 30): "; N
50 IF N < 1 OR N > 30 THEN 40
60 FOR I = 1 TO N: PRINT "LINEA
   "; I: " : "
70 INPUT T$(I): NEXT I
75 :
80 REM ELABORAZIONE
90 FOR I = 1 TO 26
100 L$ = CHR$(64 + I)
110 FOR J = 1 TO N
120 FOR K = 1 TO LEN (T$(J))
130 IF MID$( T$(J), K, 1) = L$ THEN
   L(I) = L(I) + 1
140 NEXT K, J, I
150 S = 0
160 FOR I = 1 TO 26: S = S + L(I)
   : NEXT
170 FOR I = 1 TO 26: L(I) = INT
   (L(I) / S * 10000) / 100: NEXT

175 :
180 REM VISUALIZZAZIONE
190 PRINT
200 FOR I = 1 TO 13
210 PRINT CHR$(64 + I); " : "; TAB(
   8 - LEN ( STR$( INT (L(I))
   ))); L(I);
220 PRINT TAB( 20); CHR$(77 +
   I); " "; TAB( 27 - LEN ( STR$(
   INT (L(I + 13))))); L(I + 1
   3)
230 NEXT I
250 END
```

Commenti:

Variabili:

I,J,K - contatori

N - numero di linee

S - numero totale di lettere in T\$(30)

L\$ - lettere dell'alfabeto

L(26) - tabella delle percentuali di presenza delle lettere dell'alfabeto

T\$(30) - tabella contenente le linee di testo

- 100: CHR\$(65) da la lettera A; CHR\$(64+I) da quindi la I-esima lettera dell'alfabeto.
- 90-140: con i tre cicli nidificati (linea 90 per le 26 lettere dell'alfabeto, linea 110 per tutte le linee del testo e linea 120 per ciascun carattere di ogni linea) viene cercata ogni lettera dell'alfabeto. Ogni volta che una lettera è presente viene incrementato L(I). La tabella L(I) contiene dunque il numero di volte che la I-esima lettera appare nel testo.
- 160: calcolo del numero totale di lettere.
- 170: calcolo delle percentuali arrotondate a due cifre decimali.
- 210-230: visualizzazione su due colonne delle lettere dell'alfabeto e delle percentuali corrispondenti allineate sul punto decimale.

ESERCIZIO 3.16

```
10 DIM M$(12)
20 FOR I = 1 TO 12: READ M$(I): NEXT I

30 HOME : VTAB 5
40 INPUT "GIORNO (DA 1 A 31)? ": J
   J
50 INPUT "NUMERO DEL MESE (DA 1
   A 12)? ": M
60 INPUT "ANNO? ": A
70 IF J < 1 OR J > 31 OR M < 1 OR
   M > 12 THEN 30
80 PRINT : PRINT
90 D$ = STR$(J) + " " + M$(M) +
   " " + STR$(A)
```

```

100 PRINT D$
150 END
200 DATA GENNAIO, FEBBRAIO, MARZO
    , APRILE, MAGGIO
210 DATA GIUGNO, LUGLIO, AGOSTO, S
    ETTEMBRE
220 DATA OTTOBRE, NOVEMBRE, DICEM
    BRE

```

Commenti:

Variabili:

I - contatore
J - numero del giorno
M - numero del mese
A - anno
D\$ - data
M\$(12) - i dodici mesi

20: attribuzione dei dodici mesi a M\$(12).

70: verifica della validità delle risposte.

90: creazione della data.

ESERCIZIO 3.17

```

10 HOME
20 PRINT : INPUT "DATA (ES: 11 G
    ENNAIO 1980)? ": D$
30 REM CALCOLO DEL GIORNO
40 J = VAL (D$): IF J < 32 THEN
    70
50 PRINT "ERRORE !!": GOTO 20
55 :
60 REM RICERCA DEL MESE E DELL'
    ANNO
70 FOR I = 1 TO LEN (D$): IF MID$
    (D$, I, 1) < > " " THEN NEXT
    GOTO 50

```

```

80 D$ = MID$ (D$, I + 1)
90 FOR I = 1 TO LEN (D$): IF MID$
   (D$, I, 1) < > " " THEN NEXT
   : GOTO 50
100 A = VAL ( MID$ (D$, I + 1))
110 D$ = LEFT$ (D$, I - 1)
115 :
120 REM RICERCA DEL NUMERO DEL
   MESE
130 FOR M = 1 TO 12: READ A$:
140 IF A$ < > D$ THEN NEXT : GOTO
   50
145 :
150 REM VISUALIZZAZIONE DEL RIS
   ULTATO
160 PRINT J; "/"; M; "/"; A
170 END
175 :
200 DATA GENNAIO, FEBBRAIO, MARZO
   , APRILE, MAGGIO
210 DATA GIUGNO, LUGLIO, AGOSTO, S
   ETTEMBRE
220 DATA OTTOBRE, NOVEMBRE, DICEM
   BRE

```

Commenti:

Variabili:

I - contatore
 J - numero del giorno
 M - numero del mese
 A - anno

D\$ - data
 A\$ - variabile

40: i giorni non possono superare il numero 31.

70: ricerca del primo spazio in D\$ (stesso metodo dell'Esercizio 3.5). Nel caso non si trovi alcuno spazio, il programma ritorna alla linea 50.

- 80: soppressione del numero del giorno (che è già dato da J).
- 90 : ricerca dello spazio che separa il mese dall'anno.
- 100: l'anno si trova dopo questo secondo spazio.
- 110: soppressione dell'anno, solo il mese rimane nella stringa D\$.

ESERCIZIO 4.2

```

10 ET$ = "*****"
   *****": REM 40
   STELLE
20 HOME : VTAB 5
30 INPUT "DATE UN NUMERO (TRA 1
   E 24): ";L
40 IF L < 1 OR L > 24 THEN 30
50 INPUT "DATE UN NUMERO (TRA 1
   E 39): ";C
60 IF C < 1 OR C > 39 THEN 50
65 :
70 HOME
80 VTAB L
90 PRINT LEFT$(ET$,C)
100 FOR I = 1 TO L
110 VTAB 25 - I: HTAB C
120 PRINT "*";
130 NEXT I
150 END

```

Commenti:

Variabili:

I - contatore
L,C - variabili

- 90: visualizzazione della linea di asterischi (o stelle).
- 100-130: visualizzazione della colonna di asterischi (o stelle).

Attenzione: è possibile scrivere:

```

90 FOR I = 1 TO C
93 PRINT "*";
95 NEXT I
97 PRINT

```

Non dimenticate in questo caso la seconda PRINT per poter tornare a capo sarebbe anche possibile scrivere:

```

100 FOR I = 1 TO 24 - L
110 VTAB I: HTAB C

```

ESERCIZIO 4.3

```

10 DIM N(20)
20 ET$ = "*****
          *****"
30 HOME : VTAB 5
40 REM INGRESSO DEI VALORI
50 FOR I = 1 TO 20
60 PRINT "NUMERO "; I; " (0-100)";

70 INPUT N(I)
80 IF N(I) < 0 OR N(I) > 100 THEN
60
90 NEXT I
100 REM VISUALIZZAZIONE DELL'IS
    TOGRAMMA
110 HOME
120 FOR I = 1 TO 20
130 PRINT MID$(ET$,1,N(I) / 10
    0 * 40)
140 NEXT I
150 END

```

Commenti:

Variabili:

I - contatore
ET\$ - costante

N(20) - tabella dei valori

130: visualizzazione della linea di asterischi corrispondente all'I-esimo valore.

Attenzione: non è possibile scrivere:

```
130 PRINT LEFT$(ET$,N(I)/100*40)
```

perchè questa linea provocherebbe un errore per un valore nullo di $N(i)/100*40$.

ESERCIZIO 4.4

```
10 DIM N(30)
20 HOME VTAB 5
30 INPUT "QUANTI VALORI (1-30) ?
   " : N
40 IF N < 1 OR N > 30 THEN 30
50 FOR I = 1 TO N
60 PRINT "NUMERO " : I : " (POSITIVO
   ) " :
70 INPUT N(I)
80 IF N(I) < 0 THEN 60
90 NEXT I
100 REM RICERCA DEL VALORE MASS
   IMO
110 G = 0
120 FOR I = 1 TO N
130 IF G < N(I) THEN G = N(I)
140 NEXT I
145
150 REM VISUALIZZAZIONE DELL'I
   STOGRAMMA
160 HOME
170 FOR I = 1 TO N
180 IF INT (N(I) / G * 24) = 0 THEN
   220
190 FOR J = 1 TO N(I) / G * 24
200 HTAB I : VTAB 25 - J : PRINT "
   *":
210 NEXT J
220 NEXT I
230 VTAB 24 : GET A#
250 END
```

Commenti:

Variabili:

I,J - contatori

N - numero di elementi nella tabella N(30)

G - valore più grande della tabella N(30)

N(30) - tabella di valori

110-140: stesso metodo del programma 2.6.

180 : se non si devono visualizzare asterischi, il programma prosegue alla linea 220.

190-210: visualizzazione della colonna di asterischi corrispondente all'I-esimo valore.

230: attesa della battitura di un carattere, prima della fine, perchè l'arresto del programma fa risalire le linee.

ESERCIZIO 4.6

```
5 DIM N(1,10)
10 TEXT : HOME : VTAB 5
15 REM INGRESSO DEI DATI
20 FOR I = 1 TO 10
30 PRINT "NUMERO "; I; " (PRIMO), (
    SECONDO): "
40 INPUT N(0, I), N(1, I)
50 IF N(0, I) < 0 OR N(1, I) < 0 THEN
    30
60 NEXT I
65
70 REM RICERCA DEL VALORE MAGGI
    ORE
80 G = 0
90 FOR I = 1 TO 10
100 IF N(0, I) > G THEN G = N(0, I
    )
110 IF N(1, I) > G THEN G = N(1, I
    )
120 NEXT I
125 :
130 REM VISUALIZZAZIONE DELL'IS
    TOGRAMMA
```

```

140 GR
150 FOR I = 1 TO 10
160 FOR J = 0 TO 1
165 COLOR= 7 + J
170 IF INT (N(J, I) / G * 40) >
    0 THEN VLIN 39,40 - N(J, I) /
    G * 40 AT I * 2 + J
180 NEXT J, I
200 END

```

Commenti:

Variabili:

I, J - contatori

G - il valore più grande della tabella N(1,10)

N(1,10) - tabella dei valori (N(0,x) è la prima serie di valori, N(1,x) è invece la seconda).

10 : viene dato il comando TEXT perchè lo schermo potrebbe essere in modo grafico.
30-40: ingresso degli I-esimi valori separati da una virgola.
165: colore 7 per la prima serie (J=0), colore 8 per la seconda (J=1).
170: se c'è una colonna da disegnare (INT(N(J,I)/G*40)(>0), allora bisogna tracciare una linea verticale di N(J,I)/G*40 "punti" sulla colonna I*2, per J=0, e sulla colonna I*2+1, per J=1.

Attenzione: è possibile scrivere:

```

180 NEXT J
190 NEXT I

```

ESERCIZIO 4.7

```

10 DIM V(36)
20 TEXT : HOME : VTAB 5
30 INPUT "QUANTI DADI GETTATE (D
    A 1 A 6)? ": N
40 IF N < 1 OR N > 6 THEN 20
50 HGR : HCOLOR= 3
55 :
60 REM INIZIO DEL CICLO

```

```

70 FOR I = 1 TO 800
80 S = 0
90 FOR J = 1 TO N
100 S = S + INT ( RND (1) * 6) +
    1
110 NEXT J
120 V(S) = V(S) + 1
130 IF V(S) > 159 THEN 150
140 H$PLOT S * 5, V(S)
150 NEXT I
200 END

```

Commenti:

Variabili:

I, J - contatori
N - numero dei dadi gettati
S - somma dei dadi per un lancio

V(36) - tabella delle somme (V(s) contiene il numero delle volte che la somma è stata eguale a s).

80-110: calcolo della somma del lancio di N dadi (vedere il programma 1.8).
120: incremento dell'indice della tabella che corrisponde alla somma S.
130: se una colonna esce dallo schermo, allora non viene tracciata.

ESERCIZIO 4.9

```

10 HOME : VTAB 22
20 HGR : HCOLOR= 1
30 INPUT "BATTETE LE COORDINATE
    DI UN PUNTO DELLO SCHERMO (0
    -279), (0-159) "; X, Y
40 IF X < 0 OR X > 279 OR Y < 0 OR
    Y > 159 THEN 30
50 FOR I = 1 TO 50
60 XA = RND (1) * 280 : YA = RND
    (1) * 160
70 H$PLOT X, Y TO XA, YA
80 NEXT
100 END

```

Commenti:

Variabili:

I - contatore

X,Y - coordinate del punto iniziale delle rette

XA,YA - coordinate aleatorie di un punto sullo schermo.

ESERCIZIO 4.10

```
10 HGR : HCOLOR= 3
20 FOR I = 1 TO 50
30 X1 = RND (1) * 280:Y1 = RND
   (1) * 160
40 X2 = ABS (279 - X1):Y2 = ABS
   (159 - Y1)
50 HPLLOT X1,Y1 TO X2,Y2
60 NEXT I
100 END
```

Commenti:

Variabili:

I - contatore

X1,Y1 - coordinate aleatorie di un punto sullo schermo

X2,Y2 - coordinate di un punto simmetrico a (X1,Y1) rispetto al centro dello schermo (140,80).

Attenzione: per tracciare delle rette il cui punto di mezzo sia sul centro dello schermo, è sufficiente che i punti iniziale e finale delle rette siano simmetrici rispetto al centro dello schermo (linea 40).

ESERCIZIO 4.11

```
10 TEXT : HOME VTAB 5
20 INPUT "SPAZIATURA TRA LE RIGH.
   E DELLA QUADRETTATURA (0,10)
   ":E
30 IF E < 0 OR E > 10 THEN 20
35 :
40 REM QUADRETTATURA IN GR
```

```

50 GR : COLOR= 15
60 FOR X = 0 TO 39 STEP E + 1
70 HLIN 0.39 AT X
80 VLIN 0.39 AT X
90 NEXT X
100 GET A#
105 :
110 REM QUADRETTATURA IN HGR

120 HGR : HCOLOR= 3
130 FOR X = 0 TO 279 STEP E * 5 +
1
140 HPLLOT X.0 TO X.159
150 NEXT X
160 FOR Y = 0 TO 159 STEP E * 5 +
1
170 HPLLOT 0.Y TO 279.Y
180 NEXT Y
200 END

```

Commenti:

Variabili:

OX,Y - contatori (ascisse e ordinate)

E - spaziatura

- 60: X viene incrementato per il valore della spaziatura più 1 (perchè E può essere nullo).
- 100: prima di tracciare la quadrettatura in alta risoluzione si deve premere un tasto.

ESERCIZIO 4.12

```

10 GR
20 K = 40 / 255 : C = 1
25 :
30 REM INIZIO DEL CICLO
40 COLOR= C
50 PLOT PDL (0) * K, PDL (1) *
K

```

```

60 IF PEEK ( - 16287) > 127 THEN
    GR
70 IF PEEK ( - 16286) > 127 THEN
    WAIT - 16286,128,128:C = C
    + 1: IF C > 15 THEN C = 0
80 GOTO 40

```

Commenti:

Variabili:

K - costante

C - codice del colore utilizzato

50: visualizzazione del punto le cui coordinate sono date dalle manopole dei due "paddle".

60: se è premuto il tasto del paddle 0, lo schermo grafico viene pulito.

70: se viene premuto invece il tasto 1, il programma esegue una attesa (WAIT -16286,128,128) prima di cambiare colore.

ESERCIZIO 4.13

```

10 HGR : HCOLOR= 3
20 X = 140:Y = 80
25 :
30 REM INIZIO DEL CICLO
40 HPL0T X,Y
50 GET A$
60 IF A$ > = "0" AND A$ < = "7
    " THEN HCOLOR= VAL (A$)
70 IF A$ = "R" OR A$ = "T" OR A$
    = "Y" THEN Y = Y - 1
80 IF A$ = "B" OR A$ = "V" OR A$
    = "C" THEN Y = Y + 1
90 IF A$ = "C" OR A$ = "F" OR A$
    = "R" THEN X = X - 1
100 IF A$ = "B" OR A$ = "H" OR A
    $ = "Y" THEN X = X + 1
110 IF A$ = CHR$ (13) THEN HGR
120 GOTO 40

```

Commenti:

Variabili:

X,Y - coordinate del punto visualizzato

A\$ - tasto toccato dall'utente.

ESERCIZIO 4.15

```
10 HGR : HCOLOR= 3
15 :
20 REM TRACCIAMENTO DEGLI ASSI
30 HPLOT 0,80 TO 279,80
40 HPLOT 140,0 TO 140,159
45 :
50 REM TRACCIAM. DELLA FUNZIONE

60 HPLOT 0,80 - SIN ( - 140 / 1
   0) * 30
70 FOR X = 1 TO 279
80 HPLOT TO X,80 - SIN ((X - 1
   40) / 10) * 30
90 NEXT X
100 END
```

Commenti:

Variabile:

X - contatore (ascisse del punto sullo schermo)

60: questa linea è indispensabile per avere il primo punto già tracciato nella HPLOT della linea 80.

60 e 80: ad un valore X sullo schermo corrisponde $(X-140)/10$ come ascissa della funzione. Analogamente a $F(X)$ corrisponde $80-F(X)*30$ sullo schermo.

ESERCIZIO 4.16

```
10 TEXT : HOME : VTAB 5
20 INPUT "AVETE GIA' DATO LA FUN
   ZIONE (S/N)? ";A$
30 IF LEFT$(A$,1) = "S" THEN 1
   00
40 PRINT : PRINT "ALLORA SCRIVET
   ELA NELLA FORMA : "
50 PRINT : LIST 110
60 PRINT : PRINT "BATTETE POI 'R
   UN'"
70 END
100 :
110 DEF FN Y(X) = 3 * SIN (X) +
   COS (3 * X)
120 PRINT : INPUT "ASCISSE DELL'
   INTERVALLO DI TRACCIAMENTO D
   ELLA FUNZIONE (X1,X2) : ";X1
   ,X2
130 DX = 279 / (X2 - X1):AX = -
   DX * X1: IF DX < = 0 THEN 1
   20
140 PRINT : PRINT "  CALCOLO VAL
   ORI  ": PRINT "ESTREMI DELLA
   FUNZIONE"
145 :
150 REM CALCOLO DEI VALORI ESTR
   EMI
160 Y1 = 9E37:Y2 = - 9E37
170 FOR X = 0 TO 279
180 V = FN Y(X1 + X / DX)
190 IF Y1 > V THEN Y1 = V
200 IF Y2 < V THEN Y2 = V
210 NEXT X
220 DY = 160 / (Y2 - Y1):AY = 160
   + Y1 * DY
225 :
230 HGR : HCOLOR= 3
235 :
240 REM TRACCIAMENTO DEGLI ASSI
```

```

250 IF AX > = 0 AND AX < 280 THEN
    HPLLOT AX, 0 TO AX, 159
260 IF AY > = 0 AND AY < 160 THEN
    HPLLOT 0, AY TO 279, AY
265 :
270 REM TRACCIAM. DELLA FUNZIONE

280 HPLLOT 0, AY - FN Y(X1) * DY
290 FOR X = 1 TO 279
300 HPLLOT TO X, AY - FN Y(X1 +
    X / DX) * DY
310 NEXT X
500 END

```

Commenti:

Variabili:

V - variabile

X - contatore (ascisse)

DX - intervallo asse X

DY - intervallo asse Y

AX,AY - coordinate del centro degli assi sullo schermo

X1,X2 - intervallo di analisi sull'asse X

Y1,Y2 - valori estremi delle ordinate

110: linea per la definizione della funzione.

130 : calcolo dell'intervallo sull'asse delle X e dell'ascissa dell'origine. Se l'utente ha dato un valore di X1 maggiore di X2, la funzione non puù essere tracciata.

160-210: calcolo dei valori estremi (stesso metodo del programma 2.6).

220: calcolo dell'intervallo sull'asse delle Y e dell'ordinata dell'origine.

250-260: tracciamento degli assi se non sono fuori dello schermo.

280-310: tracciamento della funzione con lo stesso metodo del programma precedente 4.15.

ESERCIZIO 4.17

```
10 GR
```

```
20 X = INT ( RND (1) * 39) + 1
```

```
30 FOR Y = 0 TO 39
```

```
40 COLOR= 15
```

```

50 PLOT X,Y
60 COLOR= 0
70 IF Y - 1 > = 0 THEN PLOT X,
  Y - 1
80 NEXT Y
90 GOTO 10

```

Commenti:

Variabili:

X,Y - coordinate della palla

20: calcolo aleatorio dell'ascisse della palla.

Attenzione: per dare l'impressione del movimento, la palla viene visualizzata una linea più in basso della sua posizione poi viene cancellata. Il processo continua sino a che la palla raggiunge il fondo dello schermo; poi la palla ritorna a cadere dall'alto.

ESERCIZIO 4.18

```

10 TEXT : HOME : VTAB 5
20 INPUT "COLORE DI FONDO (0-15)
   : ";CF
30 INPUT "COLORE DELLA BANDA (0-
   15) : ";CB
40 INPUT "LARGHEZZA DELLA BANDA
   (1-39) : ";LB
50 IF CB < 0 OR CB > 15 OR CF <
   0 OR CF > 15 OR LB < 0 OR LB
   > 39 THEN 20
55 :
60 REM DISEGNO DEL FONDO
70 GR : COLOR= CF
80 FOR I = 0 TO 39
90 ' HLIN 0,39 AT I
100 NEXT I
105 :
110 REM INIZIO DEL CICLO
120 FOR X = 0 TO 39

```

```

130 COLOR= CB
140 HLIN 0,39 AT X + LB - 40 * (
    X + LB > 39)
150 COLOR= CF
160 HLIN 0,39 AT X
170 NEXT X
180 GOTO 110

```

Commenti:

Variabili:

I - contatore

X - contatore (ascisse)

CF - colore di fondo

CB - colore della banda

LB - larghezza della banda

130-140: tracciamento di una nuova linea, nel colore della banda, sotto la banda.

150-160: cancellazione della prima linea della banda.

ESERCIZIO 4.19

```

10 TEXT : HOME : VTAB 5
20 INPUT "ALTEZZA DEL RETTANGOLO
    (1-130) : ";H
30 IF H < 1 OR H > 130 THEN 20
40 INPUT "LARGHEZZA DEL RETTANGO
    LO (1-200) : ";L
50 IF L < 1 OR L > 200 THEN 40
55 :
60 REM INIZIALIZZAZIONE
70 HGR
80 X = 0 : Y = RND (1) * (150 - H)

90 DX = RND (1) * 3 + .5 : DY = 4 -
    RND (1) * 2
95 :
100 REM INIZIO DEL CICLO

```

```

110 IF X + DX + L > 279 OR X + D
    X < 0 THEN DX = - DX
120 IF Y + DY + H > 159 OR Y + D
    Y < 0 THEN DY = - DY
130 HCOLOR= 0
140 H PLOT X,Y TO X + L,Y TO X +
    L,Y + H TO X,Y + H TO X,Y
150 HCOLOR= 3
160 X = X + DX:Y = Y + DY
170 H PLOT X,Y TO X + L,Y TO X +
    L,Y + H TO X,Y + H TO X,Y
180 GOTO 110

```

Commenti:

Variabili:

X,Y - coordinate dell'angolo alto sinistro del rettangolo

DX,DY - spostamento orizzontale e verticale

H,L - altezza e larghezza del rettangolo

80: il rettangolo parte dal bordo sinistro dello schermo da una altezza presa a caso.

90: lo spostamento orizzontale è preso aleatoriamente tra .5 e 3.5; quello verticale è compreso invece tra 2 e 4.

110-120: se uno dei bordi del rettangolo esce dallo schermo, allora si effettua un cambiamento di segno dello spostamento corrispondente.

160: calcolo delle nuove coordinate.

ESERCIZIO 5.1

```

10 TEXT : HOME : VTAB 5
20 INPUT "QUALE LIVELLO DI DIFFI
    COLTA' (0-5) ? ";N
30 GR
40 COLOR= 7
50 FOR I = 1 TO 40 + 40 * N
60 X = RND (1) * 30 + 7:Y = RND
    (1) * 40
70 PLOT X,Y
80 NEXT I

```

```

90 X = 0:Y = RND (1) * 20 + 10
100 COLOR= 15: PLOT X,Y
110 WAIT - 16287,128
120 REM INIZIO DEL GIOCO
130 DY = 0: IF PDL (0) > 170 AND
    Y < 39 THEN DY = 1
140 IF PDL (0) < 80 AND Y > =
    1 THEN DY = DY - 1
150 IF SCRN( X + 1,Y + DY) < >
    0 THEN 500
160 COLOR= 0: PLOT X,Y
170 X = X + 1:Y = Y + DY
180 COLOR= 15: PLOT X,Y
190 FOR I = 1 TO 100: NEXT I
200 IF X < 39 THEN 130
210 REM VINCITA
220 FOR I = 1 TO 500: NEXT I
230 N = N + 1: IF N < 6 THEN 30
250 FLASH : PRINT "TROPPO FORTE
    PER ME ----"
260 NORMAL : END
500 REM PERDITA
510 FOR I = 1 TO 6: PRINT CHR$
    (?): NEXT I
520 FOR I = 1 TO 3000: NEXT I
530 IF N > 0 THEN N = N - 1
540 GOTO 30

```

Commenti:

Variabili:

I - contatore

X,Y - coordinate del punto

DY - spostamento verticale

N - livello di difficoltà

40-80: lo schermo viene riempito a caso di ostacoli tra la colonna 7 e la 37. Il numero di ostacoli dipende dal livello di difficoltà N.

90-100: posizionamento iniziale a caso del punto sulla prima colonna dello schermo.

110: attesa che sia premuto il tasto del paddle 0.
 130-140: determinazione dello spostamento verticale.
 150: se il punto incontra un ostacolo, allora il programma salta alla linea 500.
 190,220,520: cicli d'attesa.

Attenzione: se il punto riesce ad arrivare al bordo destro dello schermo, allora il livello di difficoltà viene incrementato, diversamente se il punto incontra un ostacolo il livello viene diminuito.

ESERCIZIO 5.2

```

10 DIM J(2,2),N$(4)
20 HOME VTAB 5
30 INPUT "NOME DEL PRIMO GIOCATO
   RE: ";N$(0)
35 PRINT
40 INPUT "NOME DEL SECONDO GIOCA
   TORE: ";N$(1)
50 C = 0
55 :
60 REM INIZIO DEL GIOCO
70 GOSUB 1000: REM VISUALIZZAZIO
   NE DELLA TABELLA
80 F = C - 2 * INT (C / 2)
90 PRINT "   GIOCA ";N$(F)
100 INPUT "(LINEA, COLONNA)? ";LI
   ,CO
110 IF CO < 0 OR CO > 2 OR LI <
   0 OR LI > 2 THEN 100
120 IF J(LI,CO) < > 0 THEN 100
130 J(LI,CO) = F + 1
135 :
140 REM VINCITA ?
150 FOR I = 0 TO 2
160 IF J(LI,I) < > F + 1 THEN 1
   90
170 NEXT I
180 GOTO 500: REM VINCITA ?
185 :
190 FOR I = 0 TO 2

```

```

200 IF J(I,0) < > F + 1 THEN 2
    30
210 NEXT I
220 GOTO 500: REM VINCITA ?
225 :
230 FOR I = 0 TO 2
240 IF J(I,1) < > F + 1 THEN 27
    0
250 NEXT I
260 GOTO 500: REM VINCITA ?
265 :
270 FOR I = 0 TO 2
280 IF J(I,2 - I) < > F + 1 THEN
    310
290 NEXT I
300 GOTO 500: REM VINCITA ?
305 :
310 REM ALTRO GIOCATORE
320 C = C + 1: IF C < 9 THEN 60
325 GOSUB 1000
330 REM
340 PRINT : PRINT " PARTITA NUL
    LA "
350 END
495 :
500 REM VINCITA
505 GOSUB 1000
510 PRINT : PRINT "BRAVO ";N$(F)
    : " HAI VINTO! "
520 END
995 :
1000 REM SOTTOPROGRAMMA
1010 PRINT : PRINT
1020 PRINT " "; PRINT " 0
    1 2"
1021 PRINT
1030 FOR I = 0 TO 2
1040 PRINT " ";I;" ";
1050 FOR J = 0 TO 2
1060 IF J(I,J) = 0 THEN PRINT "
    ";
1070 IF J(I,J) = 1 THEN PRINT "
    X ";

```

```

1080 IF J(L,J) = 2 THEN PRINT "
      0 ";
1090 NEXT J
1100 PRINT PRINT
1110 NEXT I
1120 PRINT PRINT
1130 RETURN

```

Commenti:

Variabili:

I,J - contatori
 C - numero di mosse fatte
 F - numero della persona giocante
 CO,LI - numeri di colonna e di linea della giocata

J(2,2) - tabella che rappresenta la scacchiera (0:posizione libera, 1: posizione occupata dal primo giocatore, 2: posizione occupata dal secondo giocatore).

N\$(1) - contiene i nomi dei giocatori.

80: F vale 0 se il numero di mosse (C) è pari, vale 1 se è dispari.
 120: controllo se la posizione è già occupata.
 150-170: controllo se con la mossa appena fatta il giocatore ottiene un "filetto" sulla linea.
 190-210: stesso controllo sulla colonna.
 230-300: stesso controllo per le due diagonali.

Attenzione: per determinare se un giocatore ha vinto, è sufficiente controllare la colonna e la linea su cui ha appena giocato e le due diagonali. La vincita si ha se in uno di questi casi si trova il numero del giocatore ripetuto tre volte in fila ("filetto").

ESERCIZIO 5.3

```

10 READ M:N = RND (1) * M + 1
20 FOR I = 1 TO N: READ M$: NEXT
  I
30 T$ = LEFT$ ("-----"
  , LEN (M$))

```

```

40 C = 0
50 AL$ = "ABCDEFGHIJKLMNPOQRSTUVWXYZ"
   XYZ"
53 ET$ = "
   "

55 :
60 REM ELABORAZIONE
70 HOME : VTAB 5
80 PRINT "POTETE PROVARE "; 5 - C
   ; " VOLTE"
90 PRINT : PRINT T$
100 PRINT : PRINT "LETTERE GIA'
   PROVAE : "
110 PRINT AL$
120 PRINT ET$
125 :
130 PRINT : INPUT "QUALE LETTERA
   ? "; L$
140 FOR I = 1 TO 26
150 IF MID$(AL$, I, 1) = L$ THEN
   180
160 NEXT I
170 GOTO 130
175 :
180 REM LETTERE DELL'ALFABETO
190 ET$ = MID$(ET$, 1, I - 1) + "
   *" + MID$(ET$, I + 1)
200 F = 0: FOR I = 1 TO LEN (M$)

210 IF MID$(M$, I, 1) = L$ THEN
   F = 1:T$ = MID$(T$, 1, I - 1
   ) + L$ + MID$(T$, I + 1)
220 NEXT I
225 :
230 IF T$ = M$ THEN 500
240 IF F = 0 THEN C = C + 1
250 IF C < 6 THEN 60
255 :
260 REM PERDITA
265 PRINT : PRINT
270 PRINT : PRINT "AVETE PERSO.
   "

```

```

280 GOTO 520
495 :
500 REM VINCITA
510 PRINT : PRINT "COMPLIMENTI A
VETE TROVATO LA PAROLA! "
515 PRINT : PRINT
520 PRINT : PRINT "LA PAROLA ERA
"/"; M$; "/"
530 END
535 :
990 DATA 15
1000 DATA PAROLA, ALTRO, BRAVO, SIC
COME, APPLE
1010 DATA TAVOLO, LIBRO, DISCO, STE
LLA, QUASAR
1020 DATA CASTAGNO, MEDUSA, DISPLA
Y, SOFFITTO, APPLESOFT

```

Commenti:

Variabili:

I - contatore

M - numero di parole registrate nella DATA

N - variabile

C - numero di tentativi errati

F - indica se la lettera è contenuta nella parola (0: non è contenuta, 1: è contenuta)

M\$ - parola da cercare

T\$ - contiene tanti trattini quante sono le lettere della parola M\$. I trattini vengono sostituiti con le lettere esatte quando sono indovinate

AL\$ - lettere dell'alfabeto

ET\$ - questa stringa viene inizialmente riempita con 26 spazi. Successivamente gli spazi sono sostituiti da asterischi quando la lettera corrispondente viene provata.

- 140-170: controllo se il carattere L\$ è una lettera dell'alfabeto. Se lo è viene ricordata la sua posizione con il valore di I e il programma prosegue alla linea 180.
- 200-220: sostituzione dei trattini di T\$ con la lettera indovinata.
- 230: controllo se è stata indovinata tutta la parola.

ESERCIZIO 5.4

```

10 HOME : VTAB 5
20 PRINT "PENSATE UN NUMERO TRA
    0 E 999999, IO LO SCOPRIRÒ"

30 PRINT : PRINT
40 BI = 0:BS = 1000000:C = 0
45 :
50 REM INIZIO DEL CICLO
60 E = INT ((BS + BI) / 2)
70 PRINT "PROVIAMO CON ";E;" ?"
80 INPUT "E' TROPPO ALTO, TROPPO
    BASSO O GIUSTO (A,B,G) ? ";
    R$
90 C = C + 1
100 IF R$ = "A" THEN BS = E:GOTO
    50
110 IF R$ = "B" THEN BI = E:GOTO
    50
120 IF R$ < > "G" THEN 80
125 :
130 REM ESATTO!
140 PRINT : PRINT "EUREKA! HO IN
    DOVINATO CON ";C;" TENTATIVI
    "
150 END

```

Commenti:

Variabili:

E - numero tentativo di prova
 C - numero di prove fatte
 BI - limite inferiore

BS - limite superiore
R\$ - risposta dell'utente

120: se l'utente non da una delle tre risposte previste, allora gli viene ripetuta la domanda.

ESERCIZIO 5.5

```
10 DIM N(49)
20 HOME : VTAB 10
25 :
30 REM RIEMPIMENTO DELLA TABELLA

40 FOR I = 1 TO 49
50 N(I) = I
60 NEXT I
65 :
70 REM SCELTA DEI NUMERI
80 FOR I = 49 TO 44 STEP - 1
90 R = INT ( RND (1) * I) + 1
100 PRINT N(R); " ";
110 IF R = I THEN 150
120 FOR J = R + 1 TO I
130 N(J - 1) = N(J)
140 NEXT J
150 NEXT I
160 PRINT : PRINT
165 :
170 REM NUMERO COMPLEMENTARE
180 R = INT ( RND (1) * 43) + 1
190 PRINT N(R)
200 END
```

Commenti:

Variabili:

I,J - contatori

R - numero a caso compreso tra 1 e il numero di elementi di N(49)

N(49) - tabella

110-140: rimonta degli elementi seguenti della tabella per togliere il numero appena "uscito" (vedere il programma 2.8).

ESERCIZIO 5.6

```
10 DIM CO(51),CB(51),P$(3)
20 P$(0) = "PICCHE":P$(1) = "CUOR
   I":P$(2) = "FIORI":P$(3) = "
   QUADRI"
30 HOME : VTAB 5
40 PRINT "ATTENDETE CHE MESCOLO
   LE CARTE"
45 :
50 FOR I = 0 TO 51
60 CO(I) = I
70 NEXT I
75 :
80 REM MESCOLO DELLE CARTE
90 FOR I = 51 TO 0 STEP - 1
100 R = INT ( RND (1) * (I + 1))

110 CB(I) = CO(R)
120 IF R = I THEN 160
130 FOR J = R + 1 TO I
140 CO(J - 1) = CO(J)
150 NEXT J
160 NEXT I
165 :
170 REM VISUALIZZAZIONE
180 FOR I = 0 TO 51
190 CB = CB(I) - INT (CB(I) / 13
   ) * 13
200 IF CB < 10 THEN PRINT CB +
   1;
210 IF CB = 10 THEN PRINT "FANT
   E";
220 IF CB = 11 THEN PRINT "DONN
   A";
230 IF CB = 12 THEN PRINT "RE";

240 PRINT " DI ";
250 V = INT (CB(I) / 13)
260 PRINT P$(V),
270 NEXT I
300 END
```

Commenti:

Variabili:

I,J - contatori

R - numero a caso compreso tra 0 e il numero di elementi di CO(51)

V - seme della carta da visualizzare

CB - numero della carta da visualizzare

CO(51) - tabella delle carte ordinate

CB(51) - tabella delle carte mescolate

P\$(3) - nomi dei quattro semi

80-160: mescola delle carte riempiendo aleatoriamente la seconda tabella a partire dalla prima (vedere i programmi 2.8 e 3.11).

190: calcolo del numero della carta (0 per l'asso, 1 per il 2, ecc.) cioè il valore di CB(I) modulo 13.

250: calcolo del seme della carta.

ESERCIZIO 5.7

```
10 REM NUMERI PRIMI
20 HOME : VTAB 5
30 PRINT "2, 3, 5, 7";
40 N = 9
50 REM INIZIO DEL CICLO
60 FOR I = 3 TO SQR (N) STEP 2
70 IF N = INT (N / I) * I THEN
    100
80 NEXT I
90 PRINT ", "; N;
100 N = N + 2
110 GOTO 60
```

Commenti:

Variabili:

I - contatore

N - numero da provare se è primo

- 70: controllo se N è divisibile per I; se si allora non è primo ed il programma prosegue alla linea 100 con il numero dispari successivo.
90: viene posta una virgola tra i numeri primi da visualizzare.

ESERCIZIO 5.8

```
10 HOME
20 REM VISUALIZZ. DEL MENU' E SC
  ELTA
30 HTAB 18: PRINT "MENU'"
40 VTAB 5: PRINT "1 - ADDIZIONE"

50 PRINT "2 - SOTTRAZIONE"
60 PRINT "3 - MOLTIPLICAZIONE"
70 PRINT "4 - DIVISIONE"
80 PRINT "5 - FINE DEL PROGRAMMA"
  "
90 VTAB 12: INPUT "QUALE OPZIONE
  (1-5)? ";A
100 A = INT (A): IF A < 1 OR A >
  5 THEN 90
110 IF A = 5 THEN END
115 :
120 REM INGRESSO DEI DATI
130 HOME : VTAB 5
140 INPUT "PRIMO DATO : ";N1
150 INPUT "SECONDO DATO: ";N2
160 PRINT : PRINT
165 :
170 REM DIRAMAZIONE E VISUALIZZA
  ZIONE
180 ON A GOSUB 1000,2000,3000,40
  00
190 IF 0$ < > "" THEN PRINT N1
  ;" ";0$;" ";N2;" = ";R
200 PRINT : PRINT : PRINT "BATTE
  RE 'RETURN' PER CONTINUARE"
210 GET A$: GOTO 10
215 :
1000 REM ADDIZIONE
1010 R = N1 + N2
1020 0$ = "+"
```

```

1030 RETURN
1035 :
2000 REM SOTTRAZIONE
2010 R = N1 - N2
2020 O$ = "-"
2030 RETURN
2035 :
3000 REM MOLTIPLICAZIONE
3010 R = N1 * N2
3020 O$ = "X"
3030 RETURN
3035 :
4000 REM DIVISIONE
4005 IF N2 = 0 THEN PRINT "DIVI
      SIONE PER 0 !":O$ = "": GOTO
      4030
4010 R = N1 / N2
4020 O$ = ":"
4030 RETURN

```

Commenti:

Variabili:

A - opzione
N1 - primo operando
N2 - secondo operando
R - risultato
O\$ - segno dell'operazione

180: se A=1 allora si passa all'addizione alla linea 1000; se A=2 allora si passa alla sottrazione alla linea 2000; ecc.

190: punto di ritorno da un sottoprogramma; se O\$ <> " " allora viene visualizzato il risultato (salvo il caso di divisione per zero).

200-210: ritorno al menù dopo che l'utente ha avuto la risposta.

ESERCIZIO 6.1

Metodo 1

```
10 D$ = CHR$ (4)
20 HOME
30 PRINT "BATTETE LE PAROLE DA R
   EGISTRARE"
35 PRINT "-BATTERE 'STOP' PER TE
   RMINARE-"
40 POKE 34,3
50 PRINT D$; "OPEN PAROLE"
60 VTAB 6
65 :
70 REM ELABORAZIONE
80 INPUT P$
90 PRINT D$; "WRITE PAROLE"
100 PRINT P$
110 PRINT D$
120 IF P$ < > "STOP" THEN 30
125 .
130 REM FINE DEL PROGRAMMA
140 PRINT D$; "CLOSE"
150 TEXT : END
```

Commenti:

Variabili:

D\$ - contiene il comando 'Control-D'

P\$ - parola da registrare

90-100: scrittura della parola nel file.

110: indica al DOS che le operazioni con il file sono momentaneamente sospese e perciò la PRINT successiva visualizza sullo schermo.

Metodo 2

```
10 DIM P$(200)
20 D$ = CHR$ (4)
30 HOME
40 PRINT "BATTETE LE PAROLE DA R
   EGISTRARE"
```

```

45 PRINT "-BATTERE 'STOP' PER TE
    RMINARE-"
50 POKE 34,3
60 VTAB 6
65 :
70 REM ELABORAZIONE
80 FOR I = 1 TO 200
90 INPUT P$(I)
100 IF P$(I) = "STOP" THEN 150
110 NEXT I
120 I = 200
150 REM REGISTRAZIONE DELLE PAR
    OLE
160 PRINT D$: "OPEN PAROLE"
170 PRINT D$: "WRITE PAROLE"
180 FOR J = 1 TO I
190 PRINT P$(J)
200 NEXT J
210 PRINT D$: "CLOSE"
220 TEXT : END

```

Commenti:

Variabili:

I,J - contatori

D\$ - contiene il comando 'Control-D'

P\$(200) - tabella delle parole da registrare nel file.

ESERCIZIO 6.2

Metodo 1

```

10 D$ = CHR$(4)
20 HOME : VTAB 5
30 PRINT D$: "OPEN PAROLE"
40 PRINT D$: "READ PAROLE"
45 :
50 REM LETTURA E VISUALIZZAZIONE

```

```
60 INPUT P$
70 PRINT P$
80 IF P$ < > "STOP" THEN 60
85 :
90 PRINT D$;"CLOSE"
100 END
```

Metodo 2

```
10 DIM P$(200)
20 D$ = CHR$(4)
30 HOME : VTAB 5
40 PRINT D$;"OPEN PAROLE"
50 PRINT D$;"READ PAROLE"
55 :
60 REM ELABORAZIONE
70 FOR I = 1 TO 200
80 INPUT P$(I)
90 IF P$(I) = "STOP" THEN 150
100 NEXT I
110 I = 200
145 :
150 REM VISUALIZZAZIONE
160 PRINT D$;"CLOSE"
170 FOR J = 1 TO I
180 PRINT P$(J)
190 NEXT J
```

Commenti:

Variabili:

J - contatore

I - numero di parole nel file

D\$ - contiene il comando 'Control-D'

P\$(200) - tabella delle parole lette dal file.

ESERCIZIO 6.3

```
10 D$ = CHR$(4)
20 WT$ = D$ + "WRITE INDIRIZZI.R"

30 PRINT D$;"OPEN INDIRIZZI.L95"

40 N = 0
45 :
50 REM INGRESSO DI UN NOMINATIVO

60 HOME : VTAB 5
70 INPUT "COGNOME   ";N$
80 INPUT "NOME     ";P$
90 INPUT "INDIRIZZO ";A$
100 IF LEN(N$) + LEN(P$) + LEN
    (A$) > 90 THEN PRINT "NOMIN
    ATIVO TROPPO LUNGO! "; PRINT
    : PRINT : GOTO 70
110 HOME : VTAB 5
120 PRINT N$;" ";P$
130 PRINT : PRINT A$
140 PRINT : INPUT "CONFERMI (S/N
    )? ";R$
150 IF MID$(R$,1,1) = "N" THEN
    50
155 :
160 REM REGISTRAZIONE
170 N = N + 1
180 PRINT WT$;N
190 PRINT N$; PRINT P$; PRINT A$

200 PRINT D$
205
210 REM ALTRO NOMINATIVO
220 PRINT : PRINT
230 INPUT "VUOI DARE UN ALTRO NO
    MINATIVO (S/N)? ";R$
240 IF LEFT$(R$,1) = "S" THEN
    50
245 :
250 REM CHIUSURA DEL FILE
```

```

260 PRINT WT$: 0
270 PRINT N
280 PRINT D$: "CLOSE"
300 END

```

Commenti:

Variabili:

N - numero delle persone registrate
R\$ - variabile
D\$ - contiene il comando 'Control-D'
WT\$ - stringa contenente il comando di scrittura nel file
N\$,P\$,A\$ - cognome, nome e indirizzo della persona da registrare

100: controllo della lunghezza dei dati da registrare.
110-150: verifica dell'esattezza dei dati.
260-280: registrazione del nominativo della persona.

ESERCIZIO 6.4

```

10 D$ = CHR$ (4)
20 RD$ = D$ + "READ INDIRIZZI, R"
30 PRINT D$: "OPEN INDIRIZZI, L95"

40 PRINT RD$: 0
50 INPUT N
60 PRINT D$
65 :
70 REM ELABORAZIONE
80 HOME : VTAB 5
90 INPUT "COGNOME O NUMERO DI REGISTRAZIONE? "; F$
100 PRINT : PRINT
110 F = VAL (F$): IF F = 0 THEN
    500
120 IF F < 0 OR F > N THEN PRINT
    "IMPOSSIBILE!": PRINT : PRINT
    : GOTO 90
125 :

```

```

130 REM RICERCA PER NUMERO DI RE
    GISTRAZIONE
140 PRINT RD$:F
150 INPUT N$,P$,A$
160 PRINT D$
170 GOSUB 1000: REM VISUALIZZAZI
    ONE
180 GOTO 700: REM ALTRO NOMINATI
    VO

495
500 REM RICERCA PER COGNOME
520 FOR I = 1 TO N
530 PRINT RD$:I
540 INPUT N$,P$,A$
550 IF N$ < > F$ THEN GOTO 600

555
560 REM TROVATO!
570 PRINT D$: GOSUB 1000
580 INPUT "CONFERMI (S/N)? ";R$
590 IF LEFT$(R$,1) = "S" THEN
    700
600 NEXT I
610 PRINT D$
620 PRINT : PRINT "QUESTO NOMINA
    TIVO NON E' PRESENTE"
625 :
700 REM ALTRO NOMINATIVO
710 PRINT : PRINT
720 INPUT "VUOI CERCARE UN ALTRO
    NOMINATIVO (S/N)? ";R$
730 IF LEFT$(R$,1) = "S" THEN
    80
735 :
740 REM FINE
750 PRINT D$;"CLOSE"
760 END
995 :
1000 REM SOTTOPROGRAMMA DI
1005 REM VISUALIZZAZIONE
1010 HOME : VTAB 5
1020 PRINT N$;" ";P$

```

```
1030 PRINT : PRINT A$
1040 VTAB 12
1050 RETURN
```

Commenti:

Variabili:

I - contatore

N - numero delle persone nel file

F - numero di registrazione

R\$ - variabile

D\$ - contiene il comando 'Control-D'

RD\$ - contiene il comando di lettura dal file

F\$ - nominativo da cercare

N\$,P\$,A\$ - cognome, nome e indirizzo

40-60: lettura del numero di persone nel file.

110: se F=0 vuol dire che l'operatore ha battuto delle lettere (0 non può essere un numero di registrazione consentito) per cui la ricerca viene fatta per cognome.

560-590: conferma se la persona trovata è quella cercata (infatti più persone possono avere lo stesso cognome).



Microprocessori
e interfacciamento

Cause e effetti di una rivoluzione tecnologica

Roland Dubois Capire i Microprocessori

Dal 1972, loro data di nascita, ai giorni nostri: la fantastica rivoluzione determinata dall'invenzione del microprocessore e del microcalcolatore.

Un libro che spiega, in forma chiara e dettagliata, la funzione del microprocessore, delle memorie ROM e RAM, delle interfacce...

Con numerosi schemi di collegamento, esempi di programmi, esauriente presentazione dei principali microprocessori sul mercato.

128 pagine
Lire 10.000
Codice 342A



GRUPPO
EDITORIALI
JACKSON

Attenzione compilare per intero
la cedola
ritagliare (o fotocopiare) e spedire
in busta chiusa a:
GRUPPO EDITORIALE JACKSON
Divisione Libri
Via Rosellini, 12 - 20124 Milano

CEDOLA DI COMMISSIONE LIBRARIA

VOGLIATE SPEDIRMI

n° copie	codice	Prezzo unitario	Prezzo totale
	342A	L. 10.000	

Pagherò contrassegno al postino il prezzo indicato più L. 2000 per contributo fisso spese di spedizione.

Condizioni di pagamento con esenzione del contributo spese di spedizione.

Allego assegno della Banca

Allego fotocopia del versamento su c/c n. 11866203 a voi intestato

n° _____ Allego fotocopia di versamento su vaglia postale a voi intestato

Nome _____

Cognome _____

Via _____

Cap _____ Città _____ Prov _____

Data _____ Firma _____

Spazio riservato alle Aziende. Si richiede l'emissione di fattura

Partita I.V.A. _____

Il BASIC è ormai diventato un linguaggio internazionale parlato da tutti gli utenti e possessori di personal computer. Molte volte però non è sufficiente conoscerlo, saperlo parlare, occorre anche utilizzarlo correttamente nella stesura dei programmi. Questo libro è un validissimo aiuto per tutti quelli che si accingono a tradurre in "istruzioni" problemi di matematica, archiviazione di dati, ricerche di nominativi, disegno di grafici, di istogrammi e animazione di figure. Tutti gli esempi proposti sono presentati con molta chiarezza e di molti viene anche dato il diagramma a blocchi. Sebbene il BASIC scelto sia l'Applesoft, il libro è utilizzabile anche per altri calcolatori.

