

Programmare meglio..

SHARP PC-1500

di Fabio Marzocca

L'Input dei dati

Secondo un punto di vista molto generale, la fase di esecuzione di un generico programma può essere suddivisa in 3 classi di percorrenza ben distinte fra loro, come mostra la figura 1: fase di input, fase di elaborazione, fase di stampa o visualizzazione. Diamo ora invece un'occhiata sommaria (fig. 2) all'architettura interna di un calcolatore: in analogia con la struttura del programma, possiamo realizzare le seguenti corrispondenze:

dispositivo di ingresso - fase di input

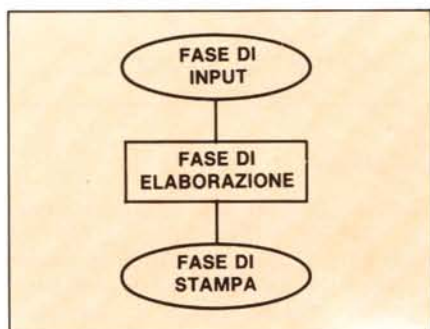


Figura 1 - Fasi di esecuzione di un programma.

CPU - fase di elaborazione
dispositivo di uscita - fase di stampa
Memoria - variabili del programma

In questo articolo vogliamo esaminare in dettaglio i vari aspetti peculiari del primo gradino del diagramma di figura 1, particolarizzato alla Sharp PC-1500: l'input dei dati.

L'istruzione INPUT

È questa senz'altro la forma più classica per l'inserimento dei dati in un programma, ma non per questo non nasconde le sue insidie. Supponiamo di avere in memoria le seguenti linee:

```
1000 I=1  
1010 INPUT @ (I): LPRINT I, @ (I):  
I=I+1: IF I <= 26 THEN 1010  
1020 END
```

La routine realizza l'input nelle memorie A-Z sotto forma vettoriale, senza peraltro rendere necessaria la dichiarazione della dimensione del vettore @ (I).

A questo punto inseriamo i dati fino all'indice I=5 e, quando si presenta il pun-

to interrogativo per la quinta volta, battiamo ENTER senza digitare alcun dato.

L'esecuzione del programma salta in questo modo alla linea 1020 trascurando completamente la linea 1010: se nella variabile @ (5) era contenuto un dato, questo non viene perduto e rimane conservato in tale cella di memoria.

Questa tecnica può essere usata agevolmente per uscire dai loop di input, magari accompagnata dal messaggio "PREMI ENTER PER CONTINUARE".

Un problema da tenere sempre a mente quando si realizzano fasi di input, è quello di considerare l'eventualità che il program-

ma venga usato da terzi. In questo caso dovrà essere rivolta particolare cura ai messaggi di commento che dovranno accompagnare ogni richiesta di dato. Vediamo questo problema con un esempio pratico.

Il breve programma di figura 6 può mostrarci la particolare caratteristica di questa singolare istruzione. Impostando sul display un numero e premendo DEF A, il programma mostrerà il valore dell'area e della circonferenza del cerchio avente per raggio quel numero; analogamente premendo DEF D avremo la conversione in dB del numero impostato, mentre con DEF C si otterrà il valore SINX/X.

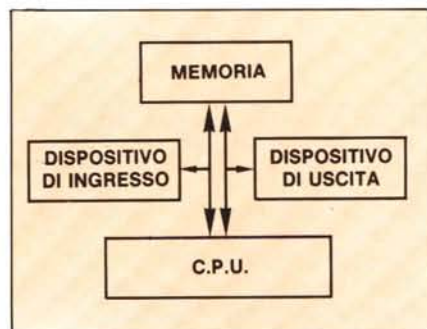


Figura 2 - Struttura generale di un calcolatore.

READ, DATA, RESTORE

Talvolta accade che alcuni dati da immettere nel programma assumano valori costanti, quali coefficienti finanziari, percentuali, diagrammi; in questi casi si consiglia l'uso delle istruzioni READ, DATA e RESTORE. La lista dei dati costanti da inserire nel programma deve seguire l'istruzione DATA secondo il seguente esempio:

```
10 DATA "ALFA", 5.3, "M", 113
```

Per leggere questi dati bisognerà realizzare una linea READ seguita dal nome

Figura 3

```
10:REM *****  
20:REM INPUT  
30:REM AUTOMATICO  
40:REM *****  
50:WAIT 0:CLS  
60:PRINT "RIGHE=  
COLONNE="
```

```
70:CURSOR 7:INPUT  
M:CURSOR 21:  
INPUT N  
80:DIM MAT(M,N):  
FOR I=1TO M:  
FOR J=1TO N  
90:W$="A("+STR$ I  
+", "+STR$ J+" )"  
=" "  
100:CLS :PRINT W$;  
110:INPUT MAT(I,J)  
:NEXT J:NEXT I
```

Figura 3 - Input automatico di una matrice.

Figura 4

```
10:REM *****  
20:REM INPUT  
30:REM MANUALE  
40:REM *****  
50:WAIT 0:CLS :  
CLEAR  
60:PRINT "RIGHE=  
COLONNE="
```

```
70:CURSOR 7:INPUT  
M:CURSOR 21:  
INPUT N  
80:DIM MAT(M,N)  
90:CLS :PRINT "A(  
, )="
```

```
100:CURSOR 2:INPUT  
I:CURSOR 5:  
INPUT J:CURSOR  
10:INPUT MAT(I  
, J)  
110:GOTO 90  
120:"A"END
```

Figura 4 - Input manuale di una matrice.

delle variabili a cui assegnare i dati, nello stesso ordine in cui sono scritti nell'istruzione DATA.

150 READ A\$, F1, C\$, F2

Per poter nuovamente utilizzare, nel corso del programma, i valori assegnati alla linea 10, bisognerà far precedere l'istruzione READ da un RESTORE con il numero di linea nella quale sono contenuti i dati:

300 RESTORE 10

310 READ C\$, GG, HS, HH

Il programma di figura 7 esemplifica l'uso combinato di queste istruzioni: si tratta di una routine che stampa il valore della componente n-esima di un vettore di lunghezza 10. Inserito il valore N, l'istruzione 520 azzerava il counter della lista DATA alla linea 500 e genera un loop di lettura da 1 a N. L'ultimo valore letto, che rappresenta la componente numero N della lista, viene visualizzato.

INPUT#

Fra le istruzioni elencate, questa è l'unica che non utilizza la tastiera per l'immissione dei dati nella memoria; INPUT# infatti è il comando che abilita l'inserimento di una lista di dati dalla porta sequenziale dell'interfaccia registratore a cassette.

Può essere impiegata sia manualmente che da programma ed in entrambi i casi bisognerà ovviamente controllare che il registratore sia posizionato nel modo "Play".

Ogni file di dati potrà essere etichettato per favorirne la ricerca:

INPUT# "DATI"; B (*)

La scrittura B (*) sta ad indicare che il computer dovrà caricare tutte le varietà di variabili "B", incluse B (1) e B stessa.

Il manuale del PC-1500 garantisce l'equivalenza delle sue istruzioni:

INPUT#, INPUT#-1,

Ciò sta ad indicare che il registratore a cassette è considerato come la periferica n° 1; è possibile infatti sostituire la cifra 1 con 2 o altre, generando strane condizioni di loop o di errore.

Questo fa quindi pensare che il PC-1500 sia internamente già predisposto per accettare nuove periferiche da cui attingere dati, o comunque, con cui dialogare con il mondo esterno.

Il caso che offre le maggiori possibilità d'errore durante la fase di input è rappresentato dal caricamento in memoria di una matrice N x M. In figura 3 è illustrato il listing di una routine denominata "Input automatico" in quanto il programma provvede automaticamente alla scansione degli indici dell'elemento (i,j): in questo caso i dati da fornire al computer devono essere inseriti per righe, cioè nella sequenza:

a₁₁, a₁₂, a₁₃ ... a_{1M}, a₂₁ ... a_{NM}

La prima richiesta di input da parte del computer è relativa alle dimensioni della matrice. Sul display apparirà:

RIGHE=? COLONNE=

impostando il numero delle righe, il punto

Figura 5

```
5000: DIM A$(0)*40
      , B$(0)*40: A$
      (0)="TECHNIM
      EDIA - VIA U
      ALSOLDA 135
      - ROMA": WAIT
      0
5010: B$(0)=A$(0)
5015: C$=INKEY$:
      IF C$=CHR$ 1
      2GOSUB 5030
5017: IF C$=CHR$ 8
      GOSUB 5040
5020: PRINT B$(0):
      GOTO 5015
5030: IF LEN B$(0)
      =1BEEP 3:
      RETURN
5035: B$(0)=RIGHT$(
      B$(0), (LEN
      B$(0)-1)):
      RETURN
5040: IF A$(0)=B$(
      0)BEEP 3:
      RETURN
5045: A=LEN A$(0):
      B=LEN B$(0):
      B$(0)=MID$(
      A$(0), (A-B),
      1)+B$(0):
      RETURN
```

Figura 5 - Routine per lo scorrimento delle scritte sul display.

Figura 6 - Routine di calcolo che utilizza l'istruzione AREAD.

Figura 7 - Esempio di impiego delle istruzioni READ, DATA, RESTORE.

interrogativo si posizionerà dopo la richiesta delle colonne. Ad esempio:

RIGHE=7 COLONNE=?

In questo modo possiamo dire che l'input è guidato da una "maschera". Successivamente, per ogni elemento, il programma mostrerà gli indici relativi al dato da inserire:

A (5,3) = ?

Il programma di figura 4 realizza invece quello che si suol dire "Input Manuale", cioè l'immissione ad accesso casuale degli elementi della matrice. Il punto interrogativo questa volta si sposterà tre volte sulla maschera di input, richiedendo rispettivamente:

1. l'indice di riga dell'elemento da inserire
2. l'indice di colonna dell'elemento da inserire
3. il valore del dato nella locazione i,j.

Per uscire dal loop di input sarà sufficiente premere DEF A. Con questa routine, inoltre, è possibile realizzare un sottoprogramma per la correzione di singoli elementi impostati errati (linee 90 - 110).

INKEY \$

È l'equivalente del GET in altri computer, e realizza l'input di qualsiasi carattere da tastiera, evitando la pressione del tasto ENTER. Va comunque ricordato che, mentre l'istruzione INPUT genera un break

Figura 6

```
10: "A"AREAD R:
      PRINT "AREA CE
      RCHIO=";PI*PI*R:
      PRINT "CIRCONF
      .=";2*PI*R:END
20: "D"AREAD A
30: D=20*LOG A:
      PRINT D;"dB":
      END
40: "C"AREAD X
50: C=(SIN X)/X:
      PRINT "(SIN X)
      /X=";C:END
```

Figura 7

```
500: DATA .5, 1.211,
      4.41, 78.5, 8.44
      , 9.1, .55, 8.71,
      14.8, 54.01
510: INPUT "Ordine
      componente? ";
      N
515: IF (N<1)OR (N>
      10)BEEP 3:GOTO
      510
520: RESTORE 500
530: FOR I=1TO N
540: READ C: NEXT I
550: W$="U("+STR$ N
      +" )="
560: PRINT W$;C:
      GOTO 500
```

alla linea di programma nella quale è inserita, INKEY \$ lascia invece il PC-1500 nello stato "Busy" senza quindi mostrare sul display il tasto premuto. Questa istruzione si presta bene come accessorio per giochi o per la gestione del display come mostra il programma di figura 5. Tale routine effettua lo scorrimento della scritta sul display in avanti o indietro premendo rispettivamente i due tasti ◀▶, e può essere molto utile nell'eventualità fosse necessario visualizzare messaggi che superano la lunghezza di 26 caratteri. I caratteri associabili all'istruzione INKEY\$ sono tutti quelli disponibili sulla tastiera del PC-1500: sul n° 11 di MC abbiamo pubblicato la lista dei codici ASCII associati ai caratteri non alfanumerici della tastiera.

AREAD

Questa è un'istruzione abbastanza insolita, presente solo nella sorella minore del PC-1500, la PC-1211; il suo significato letterale è "Automatic Reading" ed il suo impiego è legato all'uso della funzione DEF. Quando l'istruzione AREAD viene posta all'inizio di un programma etichettato, è possibile inserire un dato senza ricorrere all'INPUT. Per poter usufruire dell'AREAD è necessario però porre l'istruzione immediatamente dopo l'etichetta del programma, sulla stessa linea.

