COSTRUIAMOCI UN VERO MICROELABORATORE

HOME COMPUTER AMICO 2000

a cura della A.S.E.L. srl - parte quindicesima

Per chi ha seguito tutti gli articoli dedicati al sistema AMICO 2000 oggi non è certo un problema quello di imparare ad usare un linguaggio ad alto livello come il BASIC. L'approccio mentale acquisito nell'apprendere l'Assembler lavorando sulla piastra base è sostanzialmente lo stesso quando ci si accinge a programmare con i linguaggi dei grandi computer.

Abbiamo già accennato negli articoli precedenti che questo Mini BASIC è una versione ridotta del più potente BASIC standard di cui già oggi è disponibile la versione su ROM e su cassetta magnetica. Ad ogni modo non costituisce assolutamente una limitazione all'apprendimento il fatto che la versione ridotta del BASIC abbia un set di istruzioni ridotto. È per ciò che in questo articolo analizzeremo insieme l'utilizzo di tutte le istruzioni del Mini BASIC in modo "attivo", cioè direttamente verificandone e capendone l'uso di un programma dimostrativo creato ad hoc.

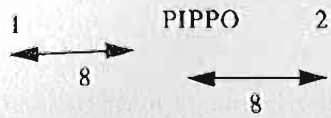
Prima di entrare nel vivo dell'argomento desideriamo chiarire alcune particolarità del nostro Mini BASIC che permettono di utilizzarlo meglio:

- Ogni linea di programma può contenere un massimo di 72 caratteri e viene "terminata" (conclusa) da un comando di carriage RETUNR (CR). Gli eventuali caatteri inseriti in più possono essere eliminati, come sapete, con il comando CTRL insieme ad H.
- Usando in modo opportuno i segni di interpunzione , e ; è possibile tabulare in modo opportuno la stampa. Utilizzando questi comandi innanzitutto si ottiene una stampa senza ritorno a capo per cui è possibile mettere su una sola riga più comandi di stampa: utilizzando il comando, (virgola) si ottiene la stampa di un numero o di un messaggio incolonnati su colonne la cui spaziatura è multipla di 8 a meno che una stringa alfanumerica non occupi più di 8 caratteri, in tal caso la prossima stringa da stampare si sposta sulla colonna multipla di 8 immediatamente successiva. Impiegando invece il comando ; (punto e virgola) si ottiene la stampa della stringa o carattere nella colonna immediatamente successiva.

Un semplice esempio chiarirà immediatamente quanto si è appena esposto. Se si batte sulla tastiera:

PRINT "1", "PIPPO", "2"

premendo RETURN otteniamo sul video:



Se il messaggio fosse più lungo di otto caratteri il secondo messaggio si sposta sulla prossima colonna di tabulazione; per esempio:

PRINT "SPERIMENTARE", "AMICO"

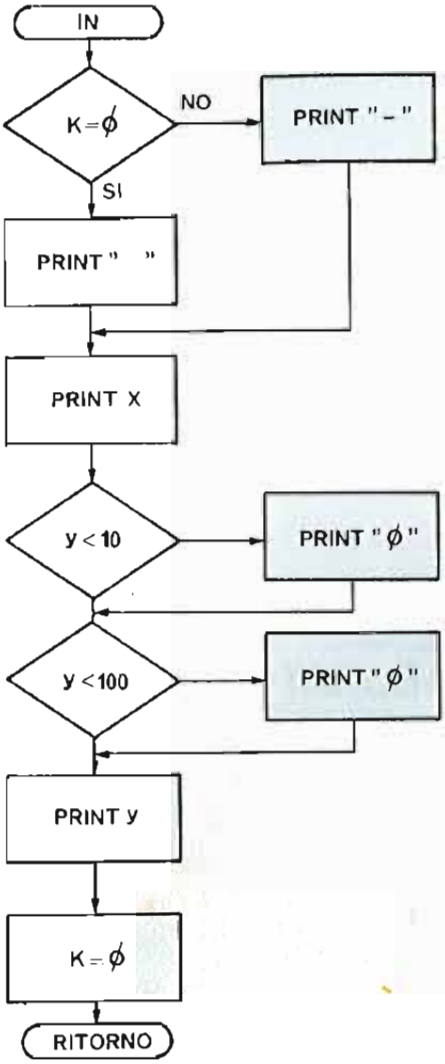


Fig. 1 - Flow-chart della routine di stampa dei numeri di 8 cifre.

premendo RETURN si ha:

SPERIMENTARE AMICO

8 8

Usando il ; le cose cambiano così:

PRINT "1" ; "PIPPO" ; "2"
premendo RETURN si ha:
1 PIPPO 2

- Spesso è possibile utilizzare lo stesso programma opportunamente sezionato per effettuare elaborazioni parziali; per far ciò basta usare l'istruzione GOTO XXXX, dove XXXX rappresenta il numero di linea da cui si intende far partire il programma.
- Per risparmiare memoria si possono scrivere le istruzioni una di seguito all'altra senza gli spazi: la riga scritta ad esempio così 510 IF A=35 GOTO 317 può essere scritta anche così 510IFA=35GOTO317; il comando di PRINT può essere abbreviato anche con PR senza mutare l'effetto della istruzione.

Visto e compreso tutto ciò passiamo subito alla descrizione del programma dimostrativo che vi aiuterà a comprendere l'uso delle istruzioni del Mini BASIC.

Il programma che abbiamo messo a punto permette di gestire matematicamente numeri di 8 cifre, mentre il Mini BASIC di per se stesso riesce a lavorare con numeri di sole 5 cifre. Questo programma in sostanza permette la somma o la differenza di numeri di 8 cifre, ma ciò che importa non è tanto quello che fa ma come e perchè lo fa. Vedremo quindi il programma nelle sue varie sottoparti o subroutine per poi, alla fine, comprendere nell'insieme come è stato pensato e scritto.

Prima di scrivere il programma si è presupposto di dividere le cifre da trattare in due parti come spiegato nell'esempio che segue: uno dei numeri da trattare è ad es. 22.418.120 esso viene ritenuto in memoria in modo che le prime cinque cifre 22418 rappresentino una variabile X e le successive tre rappresentino la variabile Y e cioè:

22.418.120
X Y

Premesso ciò cominciamo ad analizzare la routine di stampa dei numeri di 8 cifre osservando il flow-chat di Fig. 1.

Cercate di comprendere bene il funzionamento di questa parte del programma assumendo per scontato che il numero da stampare sia stato correttamente posizionato nelle variabili X e Y (sono naturalmente delle locazioni di memoria che il Mini BASIC usa per conservae numeri grandi fino a 5 cifre). I vari passaggi intermedi nella creazione del programma vengono spiegati in seguito così che alla fine ogni punto vi sarà perfettamente chiaro, così come l'intera struttura del programma.

Per quanto detto perciò la *routine di stampa dei numeri di 8 cifre* assume che la variabile X rappresenti le migliaia e la variabile Y rappesenti unità, decine e centinaia. La variabile K, che è stata opportunamente sistemata in precedenza durante l'esecuzione del calcolo, viene utilizzata per indicare il segno del numero da stampare e vale ϕ se il segno è +, mentre è diversa da ϕ se il segno è —.

Di seguito riportiamo la codifica in BASIC della routine descritta nel flow-chart di Fig. 1:

```
100 IF K=0 GOTO 103
Verifico se il numero è negativo
101 PRINT "—";
Se sì, stampo il segno —
102 GOTO 104
103 PRINT " ";
Se no, lascio uno spazio
104 PRINT X;
Stampo la parte alta del numero
105 IF Y < 10 PRINT "0"
106 IF Y < 100 PRINT "0";
Stampo gli zeri intermedi
107 PRINT Y;
Stampo la parte bassa del numero
108 LET K=0
Azzero il segno perchè sia pronto per la prossima operazione
109 RETURN
Ritorno al programma principale
```

Note: vediamo l'uso del ; che ci permette di avvicinare le varie parti del numero per scrivere le cifre di seguito. Si noti anche l'uso dell'istruzione IF non seguita dal THEN perchè il Mini BASIC interpreta tutto nel modo corretto (come se THEN ci fosse) consentendo di risparmiare preziose locazioni di memoria.

Routine di somma dei numeri di otto cifre

Questa seconda routine, il cui flow-chart è riportato in Fig. 2, assume che le

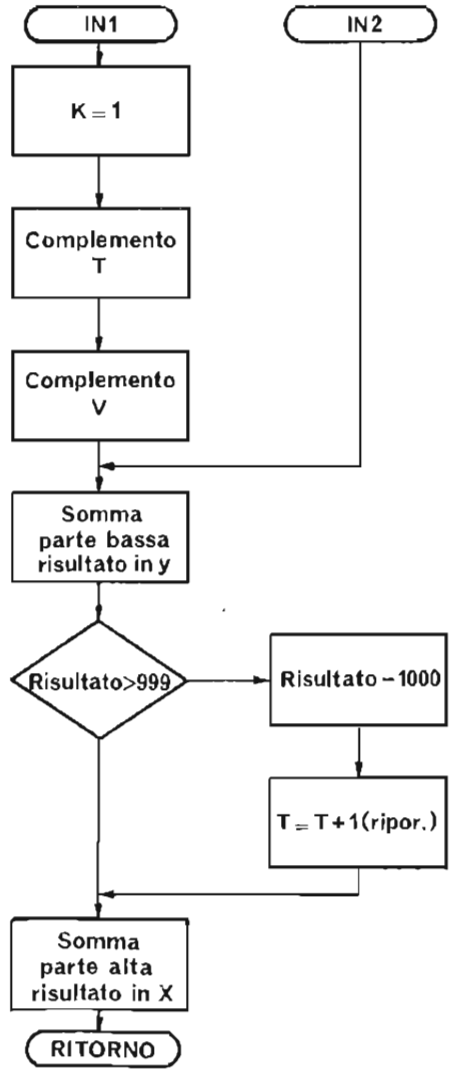


Fig. 2 - Flow-chart della routine di somma dei numeri di 8 cifre

variabili T e U rappresentino rispettivamente la parte alta e bassa del primo addendo, mentre le variabili V e Z rappresentino il secondo addendo; il risultato viene depositato in X e Y e la variabile K viene definita in base al segno del risultato.

Esaminando il flow-cart di questa routine vediamo subito che ci sono due punti di ingresso: IN1 e IN2. Una parte del programma che vedremo in seguito indirizzerà l'esecuzione della routine verso IN1 o IN2 a seconda che si tratti di sommare due numeri positivi o due numeri negativi. La differenza è importante per via del segno perchè dovrà essere negativo se i numeri sono negativi, mentre per addendi positivi il segno rimane invariato. La variabile K1 viene portata a è se il risultato è negativo, rimane a 0 se è positivo. Esaminiamo ora il listato di questa routine:

120 LET K=1
Entrata per somma di numeri negativi; metto K=1, cioè predispongo il segno — di fronte al numero
121 LET T = —T
Complemento il segno (lo porto a positivo)
122 LET T = —V
Complemento il segno (lo porto a positivo)
123 LET Y = U + Z
Sommo la parte bassa (Entrata IN2 per numeri positivi)
124 IF Y > 999 GOTO 127
Test per il riporto
125 LET X = T + V
Sommo la parte alta
126 RETURN
Ritorno
127 LET Y = Y — 1000
Sottraggo 1000 dal risultato della parte bassa per riportarla a 3 cifre
128 LET T = T + 1
Riporto (sommo) 1 sulla parte alta
129 RETURN
Rientro

Routine di differenza di due numeri di 8 cifre

Anche questa routine del programma, come quella di somma, ha due punti di ingresso che vengono selezionati da un'altra routine in base al segno dei due numeri da sottrarre.

In pratica se A e B sono i numeri da sottrarre e A - B è l'operazione da fare si entrerà in IN3 se A è positivo e B è negativo, si entrerà in IN4 se viceversa.

Dando uno sguardo al flow-chart della Fig. 3 esaminiamo ciò che accade entrando nell'ingresso IN3, essendo le procedure entrando in IN4 del tutto analoghe. La parte della routine che fa capo ai due ingressi IN3 e IN4 ha lo

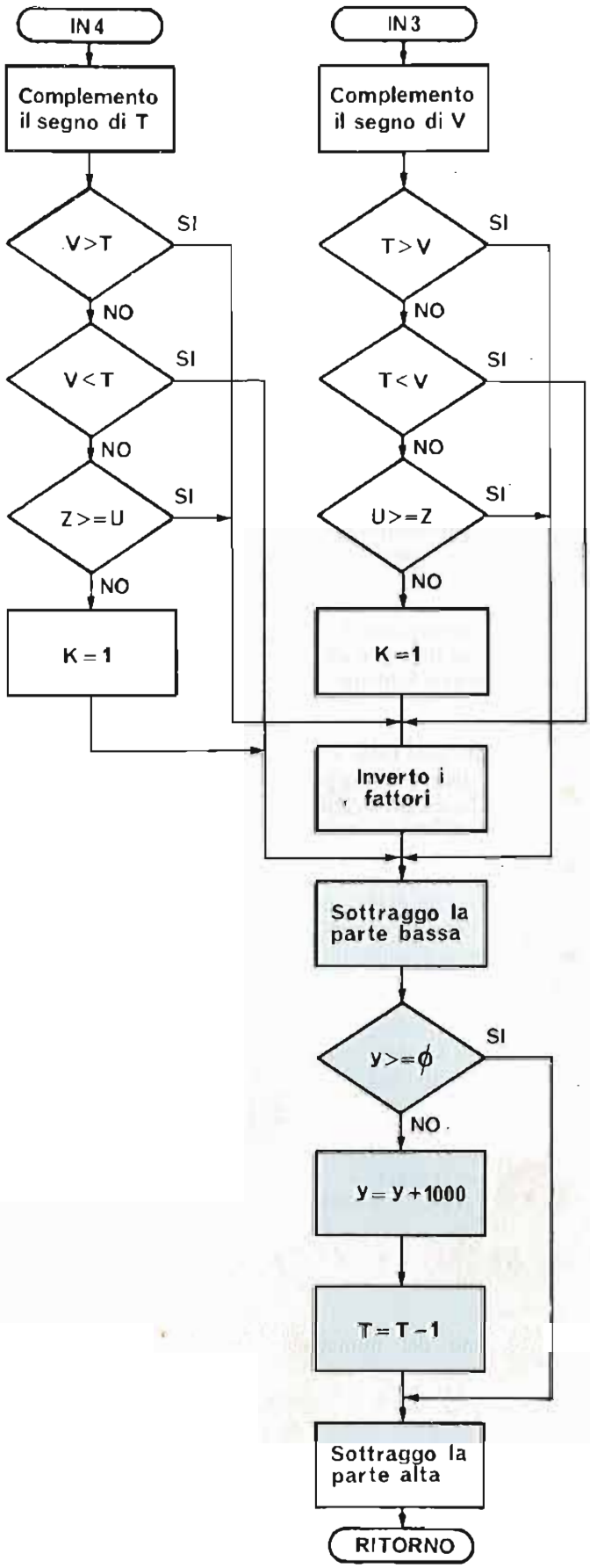


Fig. 3 - Flow-chart della routine di differenza di due numeri di 8 cifre

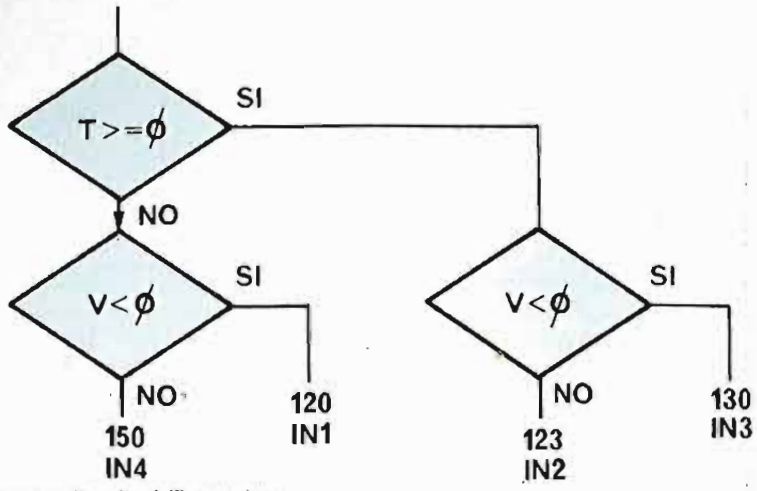


Fig. 4 - Routine di scelta dell'operazione

scopo di mettere a posto i due numeri in modo che l'operazione di sottrazione avvenga sempre fra un numero più grande e uno più piccolo.

Vediamo ora le procedure entrando in IN3: dopo aver complementato il segno del numero negativo (la parte alta V) comparo i due numeri e li metto in ordine decrescente per prepararli alla successiva operazione di differenza. Fatto ciò (analogamente vien fatto anche quando si entra in IN4) entro nella pate del programma che esegue la sottrazione: dopo aver sottratto la parte bassa dei due numeri verifico se questo risultato (Y) è positivo o negativo. Nel caso sia negativo devo effettuare una operazione di riporto. Per capire meglio questi passaggi del programma (sempre con riferimento al flow-chart) facciamo un esempio di sottrazione: 10.000 - 3.999 = sottraggo la parte bassa: 000 - 999 = — 999, il risultato è negativo per cui devo effettuare il riporto. Aggiungo allora 1.000 al risultato: — 999 + 1.000 = 1, 1 è la nuova parte bassa del numero. Una unità a questo punto deve essere sottratta ala parte alta del numero 10 — 1 = 9 e 9 — 3 = 6 per cui il risultato sarà 6001.

Di seguito riportiamo il listing di questa routine:

```
130 LET V = — V
Complemento il segno del numero
(IN3)
131 IF T > V GOTO 141
Test del segno
132 IF T < V GOTO 134
Test del segno
133 IF U > = Z GOTO 141
Test del segno
134 LET K = 1
Predispongo il segno uguale a —
135 LET X = T
136 LET T = V
137 LET V = X
Inverto i fattori
138 LET X = U
139 LET U = Z
140 LET Z = X
141 LET Y = U — Z
Sottraggo la parte bassa
142 IF Y > = ϕ GOTO 145
Test per il riporto
143 LET Y = Y + 1ØØØ
Riporto
144 LET T = T — 1
145 LET X = T — V
Sottraggo la parte alta
146 RETURN
150 LET T = — T
Complemento il segno del numero
(IN4)
151 IF V > T GOTO 135
Test per il segno
152 IF V < T GOTO 154
Test per il segno
153 IF Z > = U GOTO 135
154 LET K = 1
Predispongo il segno uguale a —
155 GOTO 141
Eseguo la differenza
```

Routine di scelta dell'operazione

Questa che analizziamo nel seguito è la routine chiave del programma perchè è quella che decide quale operazione deve essere eseguita e in quale dei quattro ingressi IN1, 2, 3, 4 deve essere indirizzato il programma (vedi il flow-chart di Fig. 4).

Questa routine è l'ultima che analizziamo e rappresenta l'inizio dell'intero programma. I dati T, U, V e Z possono essere inseriti sia direttamente da tastiera che da programma.

La routine funziona così: se T è positivo e V è positivo vado in IN1 ed eseguo una somma con risultato positivo; se T e V negativi eseguo ancora una somma con il segno del risultato negativo; se T è positivo e V è negativo vado in IN3 ed eseguo una differenza; se T è negativo e V è positivo vado in IN4 ed eseguo ancora una differenza. Per queste due ultime operazioni il segno del risultato viene determinato nell'ambito dell'esecuzione della routine di differenza.

Per queste due ultime operazioni il segno del risultato viene determinato nell'ambito dell'esecuzione dalla ruotine di differenza.

Per queste ultime operazioni il segno del risultato viene determinato nell'ambito dell'esecuzione della routine di differenza.

Riportiamo di seguito il listing di questa routine:

```
110 IF T > ϕ IF V > ϕ GOTO 123
Somma di due numeri positivi
111 IF T < = ϕ IF V < = ϕ GOTO 120
Somma di due numeri negativi
112 IF T > = ϕ IF V > = ϕ GOTO 130
Somma o differenza di due numeri di
segno opposto
113 GOTO 150
```

Come si usa il programma

Dopo aver scritto tutte le subroutine scriviamo ancora un semplicissimo programma che ci permette di trattare i dati da sommare o sottrare:

```
10 INPUT T, U
20 INPUT V, Z
30 GOSUB 110
40 GOSUB 100
50 END
```

La linea 10 di questo programma permette di inserire il 1° numero da otto cifre, la riga 20 si riferisce al 2° numero, la 30 provvede ad indirizzare ed eseguire l'operazione, la linea 40 è per la stampa dei risultati.

Facendo partire il programma (RUN più RETURN) apparirà un ?
A questo punto interrogativo si farà seguire la parte alta del numero (le prime cinque cifre), si dovrà inserire una virgola e quindi le tre cifre relative alla parte bassa del numero. Chiudiamo con un RETURN il primo numero e al ? che appare sullo schermo facciamo seguire il secondo numero scrivendolo come il primo; alla chiusura del secondo numero con il RETURN apparirà direttamente il risultato come nell'esempio che segue:

```
?12345,678 (premere RETURN)
?12345,678 (premere RETURN)
24691356
```

Con i numeri più piccoli si procede come nell'esempio che segue:

```
1221 + 2; per eseguire questa somma si
fa come segue
?1,221 (RETURN)
?0,2   (RETURN)
1223 risultato che appare sullo schermo.
```