

CONOSCERE IL COMPUTER DIRETTAMENTE DAL COMPUTER

per Commodore Vic20 e 64



Beatrice d'Este

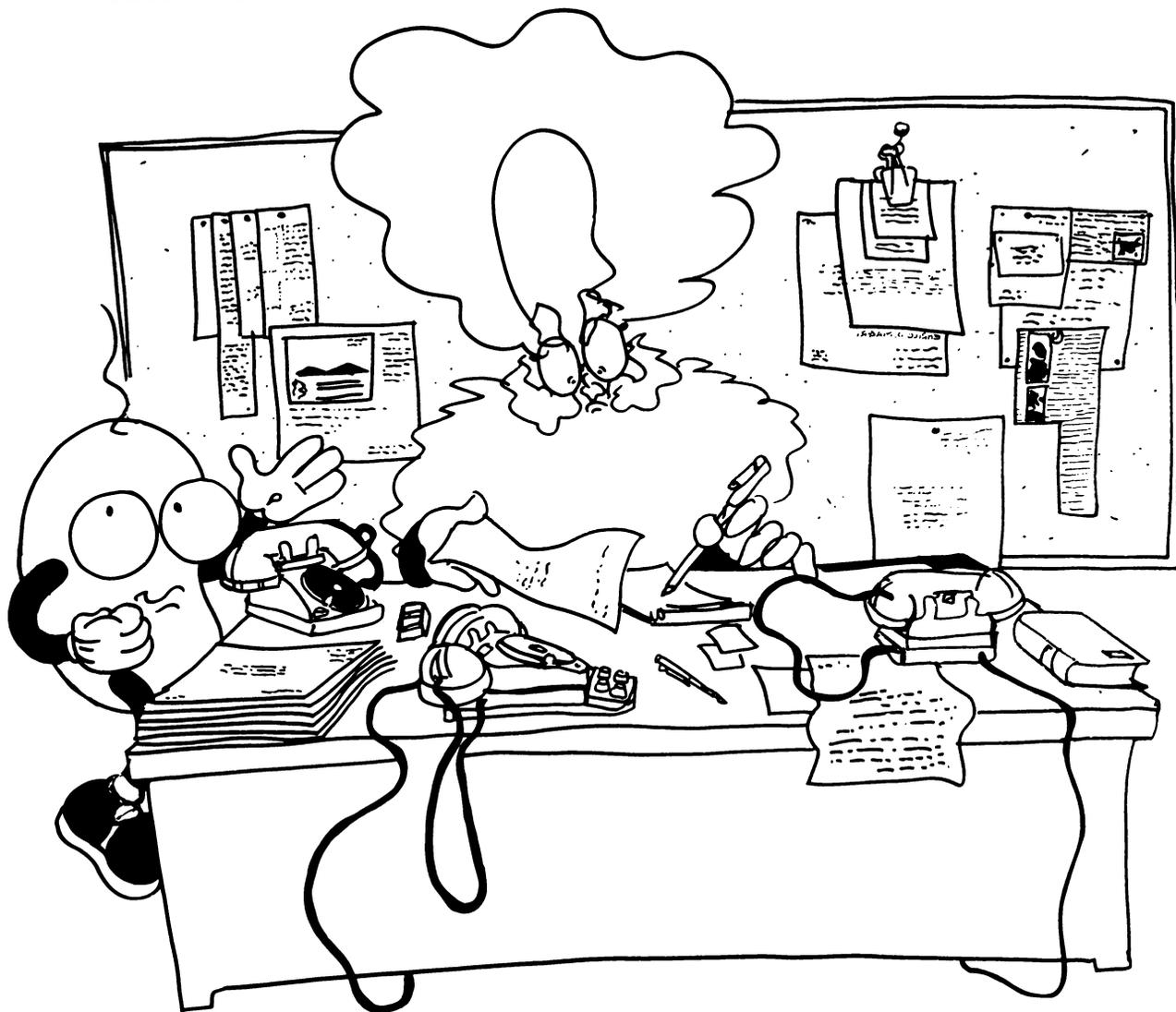
Da questa lezione vedremo le **ISTRUZIONI AVANZATE**, cioè quelle che ti permetteranno di ottimizzare i tuoi programmi rendendoli funzionali dal punto di vista della programmazione

Cominciamo con le **FUNZIONI DI STAMPA TAB, SPC** e con quelle per muovere il cursore

La stampa eseguita con il PRINT (lez 6) viene visualizzata dal punto in cui è posizionato il cursore

Ora ti mostrerò le funzioni da usare insieme al PRINT, che ti permetteranno di posizionare sul video, a tuo piacimento, le stampe dei dati in uscita.

Le posizioni possibili sono: nel CBM 64 1000, cioè 25 righe per 40 colonne, mentre nel VIC 20 506, cioè 23 righe per 22 colonne.



TAB (X);: sposta il cursore sulla colonna numero X. Quindi la prossima stampa partirà da questo punto. Ricorda che la prima colonna è numerata con lo Ø.

Cioè con: **PRINT TAB (5); "CBM"; TAB (12); "COMPUTER"**
verrà stampato: **CBM** **COMPUTER**

Ø1234**5**6789Ø1**2**3456789Ø

Nell'istruzione PRINT puoi omettere tutti i punti e virgola, tranne quello che segue l'ultimo dato da stampare.

Quindi scrivere: **PRINT TAB(5) "CBM";**

Oppure **PRINT TAB (5); "CBM";** è la stessa cosa.

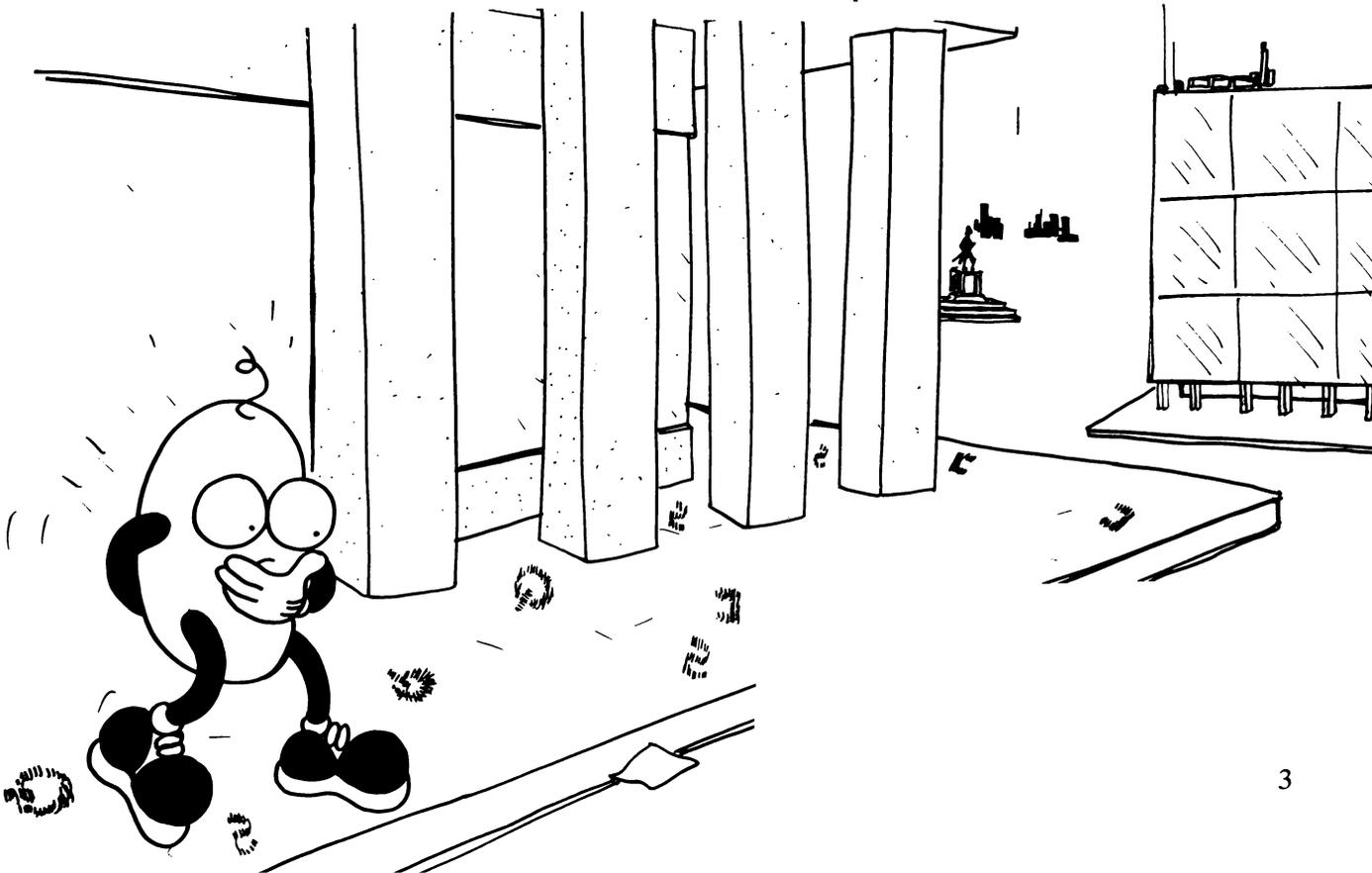
Tralasciando i punti e virgola risparmierai memoria, quindi d'ora in poi ti consiglio di ometterli sempre:

SPC(X);: sposta il cursore a destra di X spazi. Quindi la prossima stampa partirà da questo punto

Cioè con: **PRINT SPC (3) "CBM" SPC (4) "COMPUTER"**
verrà stampato: CBM COMPUTER

Ø12**3**456789Ø**4**123456789Ø

Quindi, con le funzioni TAB (X) e SPC (X) puoi posizionarti sul video, in modo da incolonnare le stampe.



Sempre con il PRINT puoi spostare il cursore nelle 4 direzioni ed eseguire le funzioni del tasto CLR/HOME.

Ma procediamo con ordine.

Per spostare il cursore dall'interno di un programma, basta stampare con un PRINT i segni degli spostamenti dello stesso cursore, compresi tra i doppi apici o memorizzati in una variabile stringa.

Per ottenere i segni degli spostamenti, dopo aver aperto i doppi apici, premi i tasti del cursore stesso.

Se stai lavorando con i caratteri minuscolo e maiuscolo, i segni appariranno così:

S	CLR
S	HOME
Q	Cursore Alto
q	Cursore Basso
I	Cursore Sinistro
J	Cursore Destro

Se stai invece lavorando con i caratteri maiuscoli e grafici, i segni appariranno così:

▼	CLR
S	HOME
○	Cursore Alto
Q	Cursore Basso
I	Cursore Sinistro
J	Cursore Destro

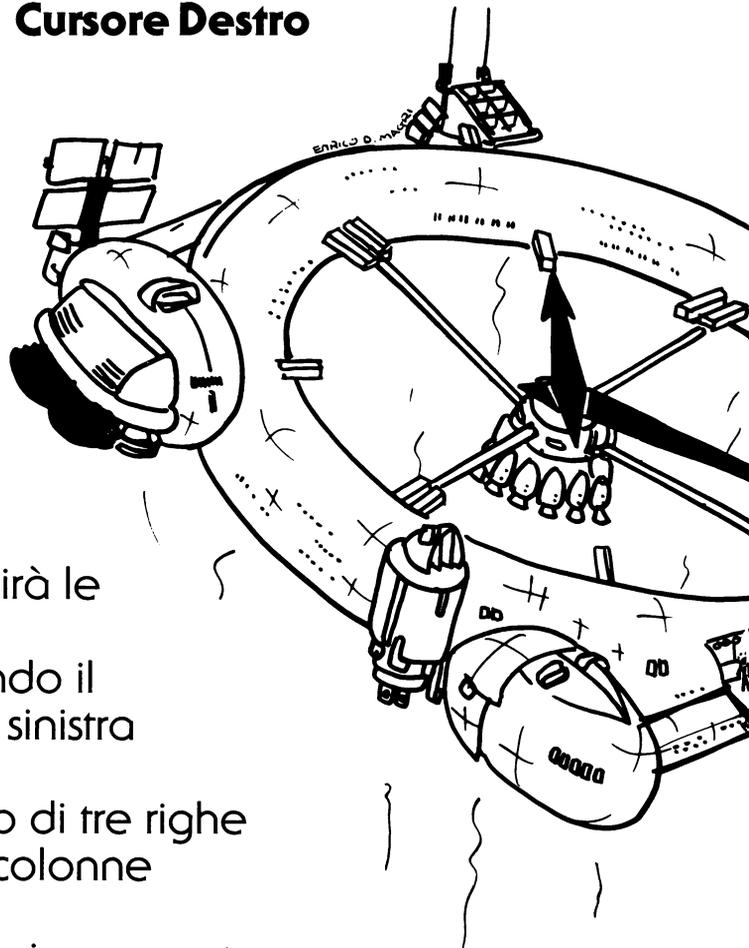
Ad esempio

PRINT "SqqqJJJCBM" eseguirà le seguenti operazioni:

1) cancellerà il video portando il cursore nell'angolo in alto a sinistra (segno **S**=CLR)

2) sposterà il cursore in basso di tre righe (segni **qqq**) e a destra di 4 colonne (segni **JJJJ**).

Da questa posizione (terza riga, quarta colonna) verrà stampata la parola CBM.



E vediamo ora le **FUNZIONI DI TEMPO**

Il computer ha infatti la possibilità di visualizzare il tempo trascorso partendo dalla sua accensione. Questo orologio, per così dire, lo puoi anche utilizzare per i tuoi programmi, usando le variabili **TI\$** e **TI**, predisposte automaticamente proprio per l'orologio.

Per leggere il tempo trascorso quindi, basterà far stampare il contenuto della variabile **TI\$** con **PRINT TI\$**

Verrà così visualizzata una stringa con 6 cifre, per indicarti le ore, i minuti ed i secondi. Ad esempio **024528** indicherà 2 ore, 45 minuti e 28 secondi.

Puoi modificare il contatore del tempo eseguendo l'assegnazione **Ti\$="OOMMSS"**, dove **OO** sta per ore, **MM** per minuti e **SS** per secondi

Ricorda che tra gli apici vanno sempre messe tutte le 6 cifre.

La variabile **TI** invece, contiene il tempo trascorso dall'accensione del computer in sessantesimi di secondo.

A questa variabile non può essere compiuta nessuna assegnazione.



Passiamo ora ad alcune istruzioni di utilizzo minore (**REM**, **CLR**, **STOP**, **CONT**), ma non per questo meno importanti. Il **REM** serve per scrivere commenti e spiegazioni dentro al listato di un programma. Tutto quello che verrà scritto dopo il **REM** non verrà visualizzato durante l'esecuzione. Il **CLR** azzerava il contenuto di tutte le variabili. Lo **STOP**, usato dentro il programma, ne ferma l'esecuzione in modo molto simile all'**END**, visualizzando però anche il numero di linea dove è avvenuta l'interruzione. Il **CONT**, usato in modo diretto (cioè fuori dal programma), permette di far ripartire l'esecuzione dal punto dell'interruzione causata dallo **STOP**. In pratica le istruzioni **STOP** e **CONT**, come vedremo più avanti, ti aiuteranno a ricercare gli errori logici dei programmi



Listato dell'esercizio: ESEMPIO PRATICO PER L'UTILIZZO DELLE FUNZIONI DI STAMPA

```
10 print" TAVOLA PITAGORICA "  
20 for x=1 to 7  
30 for y=1 to 7  
40 p=x*y  
50 p$=str$(p)  
60 printp$tab(y*3);  
70 next y  
80 print:print  
90 next x
```

Listato dell'esercizio: COMPLETA IL LISTATO

```
10 PRINT|*|(147)"SCRITTA GIREVOLE"  
20 INPUT"SCRITTA" ;|*|  
30 L=|*|(S$)  
40 'IFL>15|*|10  
50 FOR|*|=1TOL  
60 PRINTCHR$(147);S$|*|  
70 S$=RIGHT$(S$,|*|-1)+LEFT$(S$,1)  
80 FORK=1|*|200:NEXTK  
90 NEXTG  
100 FORB=1TO|*|  
110 PRINTCHR$(147)"  
120 |*|K=1TO200:NEXTK  
130 PRINTCHR$(147) ;S$  
140 FORK=1TO200:|*|  
150 NEXTB:|*|
```

PROGRAMMIAMO INSIEME

```
10 for k=1 to 5
20 input "RETURN PER TIRARE";T$
30 d1=int(rnd(0)*6+1)
40 d2=int(rnd(0)*6+1)
50 l=d1+d2
60 p=p+l
70 print:printk"TIRO"ltab(12)"PUNTI"p
200 next k
```

Soluzione dell'esercizio "COMPLETA IL LISTATO"

(apparso nella lezione n. 15)

```
10 PRINT"NUMERO INTERO";
20 INPUT X$
30 IF X$="" THEN END
40 PRINT"RISULTATO £";
50 FOR Y=1 TO LEN(X$)
60 PRINT MID$(X$,Y,1);
70 Z=LEN(X$)-Y
80 IF Z/3=INT(Z/3) THEN 110
90 IF Z=0 THEN 110
100 PRINT".";
110 NEXT Y
120 PRINT:PRINT:GOTO 10
```