

CONOSCERE IL COMPUTER DIRETTAMENTE DAL COMPUTER

per Commodore Vic20 e 64

CONOSCERE
IL COMPUTER
DIRETTAMENTE
DAL COMPUTER

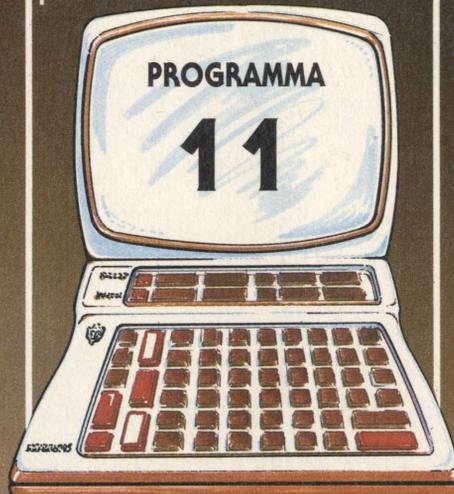
per Commodore Vic20 e 64



Beatrice d'Este

CONOSCERE
IL COMPUTER
DIRETTAMENTE
DAL COMPUTER

per Commodore Vic20 e 64



Beatrice d'Este



Beatrice d'Este



In questa lezione vediamo i **FILES SEQUENZIALI SU NASTRO**, cioè la registrazione dei dati di un programma, in modo che non vengano persi quando il computer viene spento. Questa registrazione, cioè la memorizzazione dei dati, viene chiamata **FILES** (o archivio) **DI DATI**. Naturalmente i dati del file, per poter essere nuovamente utilizzati, dovranno poi essere letti dal programma.

Le istruzioni per l'uso dei files sono:

OPEN, CLOSE, INPUT #, PRINT #, GET #.

L'OPEN serve per abilitare il collegamento tra il computer e una periferica (cioè: il registratore, l'unità a disco, la stampante, ecc) nella forma:

OPEN 1, 1, Ø, "NOME"

Dove l'OPEN è seguito dal numero del canale, cioè un numero arbitrario tra 1 e 255, dal numero della periferica, pari a 1 per il registratore, dall'indirizzo secondario, che deve essere Ø se si compie un'operazione di lettura oppure 1 se l'operazione è di scrittura, ed infine dal nome del file tra i doppi apici.



Il CLOSE serve per chiudere il collegamento aperto con l'OPEN e dovrà essere seguito dallo stesso numero di canale usato nell'OPEN. Ad esempio CLOSE 1 se l'apertura è OPEN 1.

L'INPUT# e il PRINT# devono essere usati solo dopo aver aperto il collegamento con l'OPEN e permettono rispettivamente di leggere e scrivere i dati dalla periferica in questione: in questo caso il registratore. Quindi dovranno essere seguiti dallo stesso numero di canale usato nell'OPEN.

All'INPUT# dovrà seguire una lista di variabili separate dalla virgola. Invece il PRINT# potrà anche essere seguito da costanti e espressioni, separate dalla virgola, dal punto e virgola, oppure dal CHR\$(13), corrispondente al tasto RETURN.



La virgola e il punto e virgola spaziano i dati sul nastro esattamente come nell'uso del normale PRINT (lez. 6) sul video.

La virgola aggiunge spazi tra i dati sprecando nastro, mentre il punto e virgola, non lasciando nessuna spaziatura tra i dati, ne rende impossibile la corretta lettura con l'INPUT #.

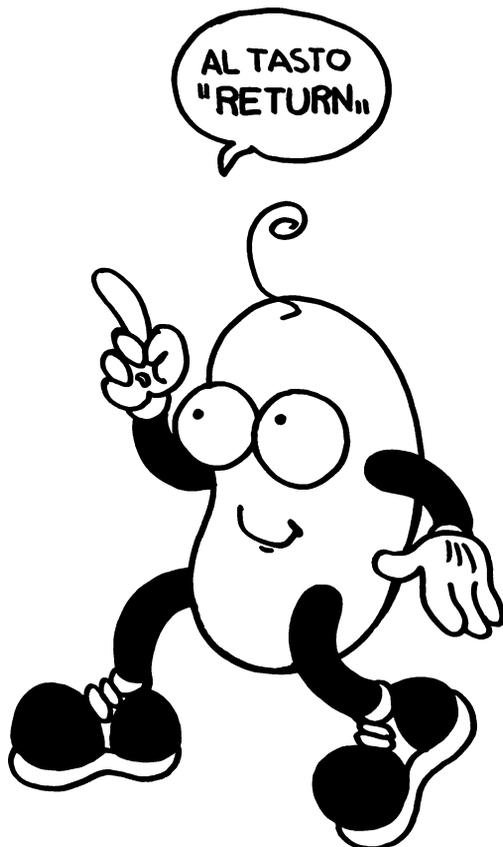
Ti consiglio quindi di dividerli con il CHR\$ (13), così ogni dato sarà separato correttamente e potrà essere riletto con l'INPUT #.

Ad esempio:

PRINT 1, A\$CHR\$ (13) 5CHR\$ (13) 6*U.

Il GET # è simile all'INPUT #, ma legge un solo carattere per volta. Naturalmente dovrà essere seguito dallo stesso numero di canale usato nell'OPEN e da una lista di variabili separate dalla virgola.

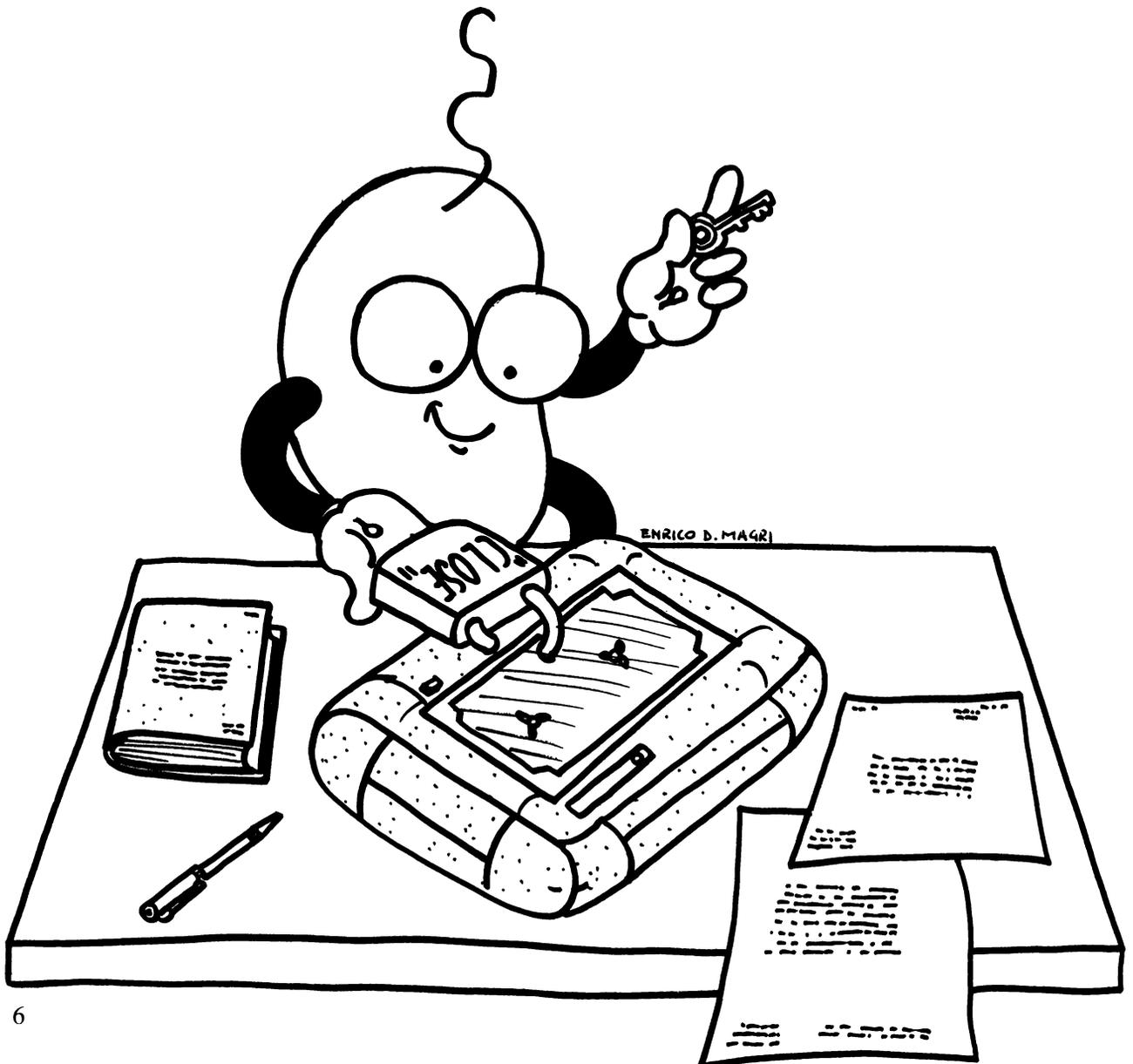
Tieni presente che la necessità di usare il GET # è molto rara.



Quindi riassumendo, il procedimento per registrare i dati sul nastro è il seguente.

Per prima cosa occorrerà aprire il collegamento con l'OPEN. Quindi con il PRINT # li scriverai sul nastro. Infine, dopo averli registrati, chiuderai il collegamento al registratore con il CLOSE.

Naturalmente ogni volta che vuoi leggere un file, la lettura partirà sempre dal primo dato. Ricorda che se leggi un dato stringa con una variabile numerica, o viceversa, si verificherà un errore.



Listato dell'esercizio: ESEMPIO PRATICO PER L'UTILIZZO DEI FILES SU NASTRO.

```
10 PRINT"REGISTRAZIONE DATI"  
20 OPEN1,1,1,"DATI"  
30 FORK=1TO9  
40 PRINTK;:INPUT"NOME";N$  
50 IFN$=""THEN40  
60 PRINT#1,N$  
70 NEXTK:CLOSE1  
80 PRINT"RIAVVOLGI IL NASTRO"  
90 PRINT"E PREMI IL RETURN"  
100 GETY$:IFY$<>CHR$(13)THEN100  
110 PRINT"LETTURA DATI"  
120 OPEN1,1,0,"DATI"  
130 FORK=1TO9  
140 INPUT#1,N$:PRINT N$  
150 NEXTK:CLOSE1
```

PROGRAMMIAMO INSIEME (CBM 64)

```
10 DIML|*|,N(14)  
20 FORK=1TO14:|*|L(K):NEXTK  
30 INPUT"IMPORTO";|*|  
40 INPUT"LIRE DATE";D  
50 IFD<ITHEN|*|  
60 |*|=D-I  
70 PRINT"RESTO L.";R  
80 FOR|*|TO14  
90 IFR>=L(K)THENR=R-|*|:N(K)=N(K)+1:GOTO90  
100 NEXTK  
110 PRINT"TAGLI DI RESTO"  
120 FORK=1|*|14  
130 IFN(K)>|*|THENPRINTN(K)"DA"L(K)  
140 |*|K  
150 DATA100000,50000,20000,10000,5000  
160 DATA2000,1000,500,200,100,50,20,10,5
```

```
10 POKE 53280,7:POKE 53281,0  
12 PRINT CHR$(147)CHR$(5)  
15 DIM L(12)  
20 FOR K=1 TO 12  
24 READ L(K)  
28 NEXT K  
30 INPUT"ANNO";Y:IF Y 1800 THEN 30  
40 INPUT"MESE";M:IF (M=0)+(M>12) THEN 40  
50 U=M  
60 IFINT(Y/4)=Y/4THENL(2)=29  
70 IFINT(Y/400)=Y/400THENL(2)=28  
80 D=1  
90 C=INT(Y/100):Y=Y-C*100  
100 M=M-2:IF M=0 THEN M=12:Y=Y-1  
110 IF M=-1 THEN M=11:Y=Y-1  
120 A=INT(2.6*M-0.1)+D+Y+INT(Y/4)+INT(C/4)-2*C  
140 X=A-7*INT(A/7)  
200 IF X=0 THEN X=7  
1000 PRINT"*****"  
1020 PRINT"LU MA ME GI VE SA DO"  
1100 FOR K=1 TO L(U)  
1120 PRINTTAB((X-1)*3)RIGHT$(" "+STR$(K),2);  
1140 IF INT(X/7)=X/7 THEN X=0:PRINT  
1160 X=X+1  
1200 NEXT K  
2000 DATA31,28,31,30,31,30,31,31,30,31,30,31
```

PROGRAMMIAMO INSIEME (VIC 20)

```
10 POKE 36879,15
12 PRINT CHR$(147)CHR$(5)
15 DIM L(12)
20 FOR K=1 TO 12
24 READ L(K)
28 NEXT K
30 INPUT"ANNO";Y:IF Y 1800 THEN 30
40 INPUT"MESE";M:IF (M=0)+(M>12) THEN 40
50 U=M
60 IFINT(Y/4)=Y/4THENL(2)=29
70 IFINT(Y/400)=Y/400THENL(2)=28
80 D=1
90 C=INT(Y/100):Y=Y-C*100
100 M=M-2:IF M=0 THEN M=12:Y=Y-1
110 IF M=-1 THEN M=11:Y=Y-1
120 A=INT(2.6*M-0.1)+D+Y+INT(Y/4)+INT(C/4)-2*C
140 X=A-7*INT(A/7)
200 IF X=0 THEN X=7
1000 PRINT"*****"
1020 PRINT"LU MA ME GI VE SA DO"
1100 FOR K=1 TO L(U)
1120 PRINTTAB((X-1)*3)RIGHT$(" "+STR$(K),2);
1140 IF INT(X/7)=X/7 THEN X=0:PRINT
1160 X=X+1
1200 NEXT K
2000 DATA31,28,31,30,31,30,31,31,30,31,30,31
```

Soluzione dell'esercizio: COMPLETA IL LISTATO (lez. n. 20)

```
10 PRINTCHR$(147)
20 PRINT
30 FORX=1TO10
40 R=INT(RND(0)*26+65)
50 A$(X)=CHR$(R)
60 NEXTX
70 S=TI:PRINT" ";
80 FORX=1TO10
90 PRINTA$(X);
100 A$=A$+A$(X)
110 NEXTX:PRINT
120 INPUTT$
130 F=TI
140 IFT$=A$THEN170
150 PRINT"HAI SBAGLIATO"
160 END
170 PRINT"SECONDI:";(F-S)/60
```