



i trucchi del CP/M

a cura di Claudio Rosazza

Maledette virgole...

L'istruzione PRINT USING, presente in numerose versioni di Basic, consente di formattare l'output di un dato inviato alla console o alla stampante. La formattazione può riguardare dati numerici o alfanumerici ed in ogni caso necessita di una stringa di mascheramento opportunamente configurata per produrre l'effetto voluto.

Nel Basic Microsoft esiste la possibilità di generare le divisioni millesimali di un dato numerico semplicemente inserendo nella stringa di mascheramento un apposito indicatore.

Ma, ahimè, il Basic Microsoft è stato realizzato negli U.S.A. e in tutti i paesi anglosassoni il significato della virgola e del punto sono rovesciati; infatti mentre noi scriviamo 1.895.445,27 gli anglosassoni scrivono 1,895,445.27.

Ora, mentre l'uso del punto come separatore decimale è ormai di uso abbastanza comune anche da noi (favorito dalla enorme diffusione delle calcolatrici portatili che usano tutte il punto decimale), vedere le virgole usate come separatori millesimali ci lascia sempre un po' perplessi.

Qui di seguito vi forniamo il modo per modificare le virgole in spazi; non è possibile modificarle in punti poiché si creerebbe un conflitto con la gestione del punto decimale; d'altronde non è consigliabile modificare il punto decimale in virgola perché ciò richiederebbe modifiche molto pesanti in tutte quelle routine interne del Basic che trattano dati numerici. Ciò non è vero per i divisori millesimali, in quanto occorre modificare un solo byte del Basic presente nella routine Print Using; vediamo come fare.

Vi sono due metodi, di cui uno modifica temporaneamente il Basic caricato in Ram e l'altro genera un nuovo Basic sul floppy.

Nel primo caso occorre caricare il Basic Microsoft in Ram ed attendere il messaggio iniziale; quindi digitate il seguente comando:

```
POKE X,&H20
```

dove X è la locazione Ram che vale &H4300 per la versione 4.51, &H352C per la 5.1 ed infine &H3557 per la 5.2.

Da questo momento il Print Using userà gli spazi come divisori millesimali, ma il Poke dovrà essere ripetuto ogni volta che si carica il Basic; per produrre una versione permanente di Basic modificato occorre usare il DDT operando come segue (in corsivo sono riportati i comandi da digitare).

Caricare contemporaneamente il DDT ed il Basic con il comando:

```
A> DDT MBASIC.COM return
```

dopo una serie di messaggi il DDT replicherà con #; digitate la sequenza di comandi:

```
#S4300 return
```

```
#4300 2C 20 return
```

```
#4301 23 . return
```

```
#G0 return
```

```
A> SAVE YY EBASIC.COM return
```

L'esempio è dato per il 4.51, per le altre versioni occorre sostituire 4300 con le locazioni di cui sopra (352C o 3557). L'indicatore YY dell'ultima riga riguarda il numero di pagine Ram da salvare su disco ed è pari a 72 per il 4.51 ed a 95 per le versioni 5.1 e 5.2. Il File Ebasic è ora il Basic modificato, mentre il file Mbasic continua ad essere la versione originale. Qualche problema in più sorge se il programma viene compilato con il Bascom; in questo caso infatti l'interprete non ha più peso poiché sostituito da moduli prelevati dalla libreria e facenti parte del programma stesso che ora non è più un .BAS ma un .COM. Inoltre la posizione del byte da modificare non è fissa, ma dipende dalla lunghezza del programma e dal tipo di istruzioni; occorre pertanto andarsi a "cercare" il byte interessato sul programma .COM. Tale funzione è realizzata dal programma ECONV.BAS

listato qui di seguito che scandisce il file .COM alla ricerca del famigerato byte, sostituendolo. Il programma richiede unicamente il nome del file da convertire che deve essere un .COM ed un prodotto della compilazione e linking con il Bascom Micro-soft.

Il programma produce una modifica permanente ed è compatibile per tutte le versioni di Bascom dalla 5.0 alla 5.2X; per riavere la versione originale del vostro programma occorre ricompilarlo.

Sul Bascom 5.3 la situazione si semplifica poiché la routine di Print Using non viene incorporata nel programma .COM, ma fa parte della libreria Run-Time BRUN.COM. In questo caso è sufficiente convertire una volta per tutte BRUN.COM e da quel momento ogni compilazione produrrà un programma con il Print Using modificato. Non essendo prevista la possibilità di riportare il BRUN.COM allo stato originale né la possibilità di modificarne il nome, vi consigliamo di fare più di una copia del file originale su floppy diversi.

```
10 DEFINT I, J: Y$=CHR$(&HC0)+"E, "+CHR$(&H23)
20 FOR I=1 TO 24:PRINT:FOR J=1 TO 90:NEXT J, I
30 PRINT "ECONV European Bascom Converter"
35 PRINT:PRINT:PRINT
40 INPUT "File da convertire ", F$
50 F$=F$+".COM"
60 OPEN "R", 1, F$
70 FIELD #1, 128 AS A$
80 FOR I=1 TO 512
90 GET #1, I
100 IF LEFT$(A$, 1)=CHR$(2E) THEN 120
110 IF EOF(1) THEN 270
120 J=INSTR(A$, Y$)
130 IF J<>0 THEN 200
140 IF MID$(A$, 126, 3)=CHR$(&HC0)+"E, " THEN 300
150 IF MID$(A$, 127, 2)=CHR$(&HC0)+"E" THEN 360
160 IF MID$(A$, 128, 1)=CHR$(&HC0) THEN 420
170 NEXT I
180 CLOSE
190 GOTO 40
200 A1$=MID$(A$, 1, J+1)
210 A2$=" "+MID$(A$, J+3, 129-J+3)
220 LSET A$=A1$+A2$
230 PUT #1, I
240 PRINT TAB(31)"Conversione effettuata":PRINT
250 CLOSE
260 GOTO 40
270 PRINT TAB(31)"File già convertito o senza
l'uso di Print Using":PRINT
280 CLOSE
290 GOTO 40
300 B$=A$
310 GET #1, I+1
320 IF MID$(A$, 1, 1)(<)CHR$(&H23) THEN 170
330 A1$=MID$(B$, 1, 127)
340 LSET A$=A1$+" "
350 GOTO 230
360 GET #1, I+1
370 IF MID$(A$, 1, 2)(<) ", "+CHR$(&H23) THEN 170
380 A1$=MID$(A$, 2, 127)
390 LSET A$=" "+A1$
400 I=I+1
410 GOTO 230
420 GET #1, I+1
430 IF MID$(A$, 1, 3)(<) "E, "+CHR$(&H23) THEN 170
440 A1$=MID$(A$, 3, 126)
450 LSET A$="E "+A1$
460 I=I+1
470 GOTO 230
```

Listing del programma da usare per eseguire la modifica nel caso di programmi compilati.