



I codici di errore dello ZX81

di Fabio Berno

0/

0/ Tutto è OK, il programma è terminato normalmente senza errori.

1/

1/ Si è fatto il NEXT di una variabile non inizializzata con un FOR, ma la variabile esiste già; quindi la responsabile dell'errore è una variabile con un nome lungo 1 solo carattere, cosa da evitare sullo ZX81, se possibile, proprio perchè si può fare confusione con i FOR... NEXT. Ad esempio, se ho dichiarato

```
10 LET A=10
```

e poi faccio

```
3333 FOR A=-7 TO 93
```

il contenuto di A verrà modificato durante il ciclo di FOR.

Per risolvere la situazione, bisogna vedere alla linea errata quale è la variabile usata, sostituirla se necessario, e inserire a monte dell'errore la linea con il FOR.

2/

2/ Uso di una variabile non definita in precedenza. Bisognerà quindi inserire, a monte dell'errore, la dichiarazione della variabile: ciò può avvenire con un LET, un DIM, un FOR (l'errore 2, al po-

sto dell'errore 1, compare nei cicli di FOR se la variabile usata col NEXT non esiste).

3/

3/ Indice di variabili indicizzate (come gli array numerici o di caratteri) fuori dai limiti stabiliti. Ciò avviene generalmente quando si indirizzano elementi di array tramite variabili che vengono alterate da cicli FOR... NEXT, o che vengono modificate impropriamente.

Controllare quindi manualmente passo per passo lo svolgimento del programma nella "zona" dell'errore, fino a trovare cosa fa "sballare" lo ZX81.

NOTA: se l'indice ha valore negativo o è maggiore di 65535 questo discorso resta valido, ma il computer segnala errore "B".

4/

4/ Non c'è più spazio in memoria. Durante l'esecuzione di un programma ciò può avvenire se, ad esempio, si è dimensionato un array che occupa troppo spazio.

L'unica soluzione è dare un bel CLEAR, cancellando così tutte le variabili, e cercare di analizzare il programma per risparmiare spazio il più possibile, specialmente se si lavora (ancora!) con 1K di memoria. Ad esempio, si possono togliere dal programma le linee con i

```
de)=48 AND code<=57 AND x<=24 OR  
code)=97 AND code<=102 OR code=  
32 THEN POKE 23658,0: PRINT BR  
IGHT 1: PAPER attr:AT y,x:INKEY$  
: PAUSE pa: POKE 23658,0: IF x<  
30 THEN LET x=x+1: GO SUB 6100  
160 IF CODE INKEY$=13 THEN GO  
SUB 8000  
170 IF CODE INKEY$=204 THEN LE  
T ns=ns+132: GO SUB 4000  
180 IF CODE INKEY$=205 THEN LE  
T ns=ns-132: GO SUB 4000  
190 IF CODE INKEY$=198 THEN IN  
PUT "SAVE Start ";a$: GO SUB 300  
0: LET ss=num: INPUT "SAVE End";  
a$: GO SUB 3000: LET ss=num: INP  
UT "TITOLO";a$: SAVE a$CODE ss,s  
e=ss  
200 IF CODE INKEY$=197 THEN IN  
PUT "Numero da Convertire";a$: G  
O SUB 3500  
500 GO TO 90  
2999 REM input esad dec  
3000 IF a$(1 TO 1)="#" THEN GO  
TO 3020  
3010 LET num=VAL a$: RETURN  
3020 LET n$a$(2 TO 5): GO SUB 6  
040: LET num=dec: RETURN  
3499 REM conversione  
3500 IF a$(1 TO 1)="#" THEN GO  
TO 3520  
3510 LET n$=VAL a$: GO SUB 5050:  
OPEN "a$.txt" FOR APPEND: PAUSE  
AT 0: PRINT n$: GO TO 3500  
3520 LET n$a$(2 TO 5): GO SUB 6  
040: LET num=dec: RETURN  
3800 REM conversione  
4010 LET n$a$(2 TO 5): GO SUB 6  
040: LET num=dec: RETURN  
4020 LET n$=VAL a$: GO SUB 5045:  
PRINT PAPER 5:AT v,0:r$  
4030 FOR o=6 TO 21 STEP 3  
4040 LET dec=PEEK ((n+v*6)+(o/  
)-2): LET t=dec: GO SUB 5000  
INT AT v,0:r$  
4045 IF t>164 n  
t=63  
4047
```



REM, o diminuire il numero delle scritte non necessarie sullo schermo, oppure ancora usare sempre le stesse variabili.

Certamente, così risparmiare memoria diventa un'arte, ma è una cosa essenziale su qualunque computer ridurre al minimo lo spazio occupato dal programma e dai dati.

5/

5/ Non c'è più spazio sullo schermo. Causa: PRINT o LIST. Soluzione: fare attenzione nel gestire le scritte sullo schermo, inserendo degli opportuni CLS, oppure usando lo SCROLL, oppure usando il seguente trucchetto, (se si ha bisogno di un output organizzato a pagine):

```
200 IF PEEK 16442 >= 4 THEN
GOTO 230
210 PAUSE 33333
220 CLS
230.....
```

così, quando lo schermo sarà pieno, (PEEK 16442 è il numero di linea su cui opera PRINT) il programma si fermerà e potrà ripartire solo premendo un tasto (NON BREAK!)

NOTA: CLS è solo indicativo, a seconda dei casi si userà un COPY, o altro.

6/

6/ Overflow aritmetico, si è cioè andati al di là del massimo numero consentito sul calcolatore (10^{38}).

Visto che di solito ciò avviene nella valutazione di funzioni matematiche, (esempio: TAN PI/2) bisogna fare molta attenzione nell'usarle.

7/

7/ Linea in cui c'è un RETURN a cui non corrisponde nessuna chiamata di GOSUB.

Probabilmente il caso più comune è quello di una subroutine in fondo al programma a cui si accede perché non si è messo uno STOP prima dell'inizio della subroutine, come in questo esempio:

```
100 GOSUB 300
110
120
```

```
290 REM QUI CI VORREBBE UNO
STOP O UN GOTO
300 REM SUBROUTINE
```

350 RETURN

Qui il computer entra nella subroutine credendola una normale sezione di programma, e quando trova il RETURN si blocca, appunto con errore 7. Prima del corpo della subroutine bisogna quindi trasferire il controllo altrove.

8/

8/ Errore abbastanza raro: viene segnalato solo quando si tenta di fare un INPUT in modo immediato (cioè da tastiera). Dando, ad esempio, INPUT ABC, il computer segnalerà appunto 8/0: L'unico modo di evitarlo è quello di... evitare di dare un INPUT in modo immediato.

9/

9/ Il programma si è fermato a uno STOP. Dando CONT in modo immediato il programma ripartirà dalla linea successiva a quella dello STOP. Questo non è, quindi, un vero e proprio errore, ma solo l'avvertimento che il programma si è fermato per nostra volontà, invece che alla sua fine logica o per errore.

A/

A/ Argomento non valido per certe funzioni matematiche, come: SQR e LN (argomento negativo); ASN e ACS (valore minore di -1 o maggiore di 1).

B/

B/ Se l'errore è segnalato in una linea che usa un array, vale il discorso fatto per l'errore 3, con l'eccezione segnalata nella nota.

Se invece non è così, allora nella linea errata si ha un intero o una variabile fuori dai limiti dell'istruzione in cui è usata, come nei seguenti casi:

```
RUN, GOTO, GOSUB, LIST, LLIST n
con n>9999;
DIM A(N), RAND, PEEK, PAUSE n
con n>65535;
POKE N,1 con N>65535 o i>255;
CHR$ X con X>255;
PLOT, UNPLOT X,Y con X>63 o
Y>43;
```

Per tutti gli esempi fatti l'errore si ha anche se l'argomento è negativo.

C/

C/ Il testo della stringa di cui si cerca di fare la VAL non costituisce un'espressione aritmetica valida, ciò succede se nella stringa sono contenute variabili non dichiarate o se c'è qualche errore sintattico-aritmetico (esempio: VAL "A+=3").

D/

D/ Il programma è stato interrotto con un BREAK da tastiera; può essere dato quindi in ogni caso: alla fine dell'esecuzione di una linea di programma come in un LOAD, SAVE, LPRINT, LLIST, COPY.

Oppure si è risposto con STOP a una richiesta di INPUT numerico (è quindi un modo per fermare un programma).

E/

E/ Indice non usato.

F/

F/ Si è fatto un SAVE"" (stringa vuota), che a differenza di LOAD non è permesso.

Nota: Il numero che compare sul video dopo il codice di errore e lo slash (il segno /) indica la linea dove è avvenuta l'interruzione.

In un prossimo numero illustreremo i codici di errore dello Spectrum.

