a cura di Francesco Petroni

Archivi per dati di grafica edile e architettonica - 2ª parte

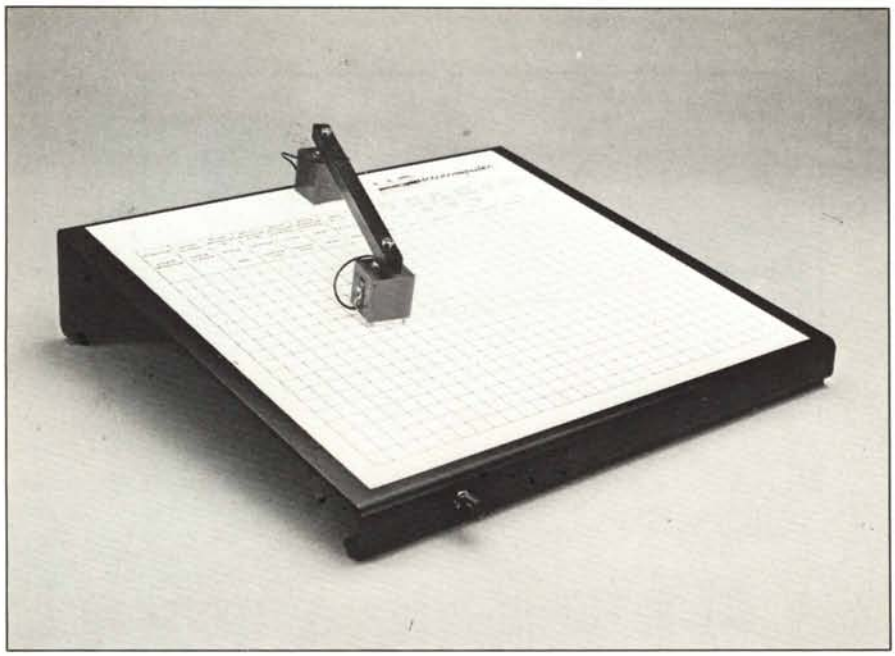

Figura 1 - La tavoletta grafica di MC; il programma utilizza per l'input dei dati la tavoletta grafica di MC, nel formato standard di 280 per 182 punti.

Nell'articolo pubblicato il mese scorso abbiamo proposto un sistema di codifica per archivi di dati grafici utile in Computer Grafica applicata al disegno edile e architettonico.

Siamo partiti dalla fase di OUTPUT, ovvero inseriti a mano dati grafici già codificati, li abbiamo elaborati e visualizzati con un programma chiamato DISEGNO.

In questo numero tratteremo la fase di INPUT degli stessi dati, realizzata utilizzando la tavoletta grafica di MC collegata ad un Apple II.

Chi possiede una tavoletta grafica di MCmicrocomputer ne ha apprezzato il potente software applicativo che permette facilmente la realizzazione di disegni anche molto complessi sul monitor dell'APPLE II e la memorizzazione in PICTURE.

Ovvero il disegno può essere memorizzato tutto insieme su dischetto e può essere richiamato tutto insieme per la visualizzazione.

Il nostro obiettivo è invece quello di codificare opportunamente gli elementi costitutivi del disegno e memorizzarli non più come PICTURE ma come dati inseriti in un vettore, trattabile quindi come un file. Su questi dati sarà possibile fare tutte le elaborazioni e le visualizzazioni che si vuole.

Nel realizzare il nostro programma IMMISSIONE ci siamo posti alcune condizioni iniziali che è opportuno dichiarare subito:

- l'input dei dati deve avvenire tramite tavoletta grafica di MC, opportunamente calibrata con il programma di Calibrazione fornito con il software di corredo;

```
     FUNZIONE DI CALIBRAZIONE

 PUNTA IN ALTO A SINISTRA

VALORE X AZZERAMENTO -144
VALORE Y AZZERAMENTO 60
```

Figura 2 - La fase di calibrazione; la fase di calibrazione serve per determinare le posizioni reciproche tra zona menu e zona di input vero e proprio.

- il menu del programma occupa la stessa posizione che occupa fisicamente il menu sul ripiano plastificato della tavoletta, individua cioè 42 funzioni differenti (3 righe da 14);
- il programma deve essere modulare, ovvero ogni utilizzatore può inserire a suo piacimento funzioni semplicemente inserendo una subroutine e rispettando qualche condizione al contorno;
- la fase di input può essere seguita sia sulla pagina TEXT che su quella HGR2. Deve quindi essere possibile passare da una pagina all'altra senza perdere informazioni;
- in caso di errori di immissione il programma deve permettere correzioni, senza cioè dover iniziare tutto da capo.

Insomma si è studiata una architettura modulare, facilmente implementabile e modificabile. Per cui sconsigliamo di copiare il programma così come è, cosa che tra l'altro risulterebbe molto noiosa, senza averlo preventivamente capito.

Sarà possibile costruire il programma pezzo per pezzo, inserendo solo i moduli che interessano, creandone dei nuovi, modificandone di esistenti, ecc. Oltrettutto la costruzione passo passo permette pure la prova passo passo delle singole funzioni.

Architettura generale

Le 42 funzioni individuate dal menu della tavoletta sono rispettivamente in riga 100, 200, 300,.... 4.100, 4.200.

Nel programma ne sono implementate solo 14 le altre 28 rimangono a vostra disposizione.

Le 42 caselle sono state suddivise secondo una semplice logica: le prime 14 contengono funzioni di utilità, che non generano un codice da memorizzare nel vettore dati, né generano dati, le altre 28 invece generano un codice (noi abbiamo messo i numeri 1001, 1002, 1003,.... 1028), che viene memorizzato insieme ai dati necessari per realizzare quella funzione.

Oltre a queste subroutine, ciascuna individuante una funzione, che costituiscono la parte "secondaria" del programma c'è quella che potremo definire l'ossatura principale, ovvero le varie routine che permettono via via lo svolgimento del programma con l'entrata e l'uscita dalle varie routine.

Il programma inizia (riga 10) con la pulizia delle pagine e con il richiamo delle due routine di inizializzazione.

La routine a riga 5.000 contiene il caricamento del PADDLE.CODE necessario alla lettura della tavoletta. Poi contiene la definizione delle funzioni necessarie alla lettura delle posizioni di memoria contenenti i dati paddle e alla loro traduzione, tramite formule trigonometriche, in coordinate XY.

C'è poi il caricamento dei nomi delle 42 funzioni inserite. Si può stampare l'elenco dei codici, con il progressivo, codice e descrizione, togliendo l'istruzione di salto di riga 5140. Il risultato è in figura 2.

La routine di riga 5.500 invece permette la taratura della tavoletta. Ovvero va puntato il vertice superiore sinistro della zona INPUT che è coincidente con il vertice inferiore sinistro del menu. Questo è il punto origine (cioè di coordinate 0,0) del nostro riferimento, che in realtà avrebbe avuto coordinate X0%, Y0%.

I valori necessari per la traduzione delle coordinate successivamente immesse vengono visualizzati (vedi fig. 3).

```
N.   CODICE   FUNZIONE

1             SWITCH
2             CLEAR
3             CORR. ULTIMA FUNCT.
4             FINE PROGRAMMA
5             F5
6             F6
7             F7
8             F8
9             F9
10            F10
11            F11
12            SALV. DISCO PICTURE
13            SALV. DISCO FILE
14            STAMPA DATI
15   1001     PUNTO
16   1002     SEGMENTO
17   1003     SPEZZ. QUALS.
18   1004     SPEZZ. AP. OR. VERT.
19   1005     SPEZZ. CHI. OR. VERT.
20   1006     RETTANG.
21   1007     CERCHIO
22   1008     F22
23   1009     F23
24   1010     F24
25   1011     F25
26   1012     F26
27   1013     F27
28   1014     F28
29   1015     F29
30   1016     F30
31   1017     F31
32   1018     F32
33   1019     F33
34   1020     F34
35   1021     F35
36   1022     F36
37   1023     F37
38   1024     F38
39   1025     F39
40   1026     F40
41   1027     F41
42   1028     F42
```

Figura 3 - Elenco dei codici e delle funzioni; le prime 14 funzioni non provocano l'identificazione di un codice. Dalla 15.ma invece il codice identifica la funzione e quindi identifica il significato dei dati che seguono.

A questo punto le coordinate puntate avranno valori $0 \leqslant X \leqslant 279$ e $0 \leqslant Y \leqslant 191$ nella zona INPUT, e valori $0 \leqslant X \leqslant 279$ e $-40 \leqslant Y < 0$ nella zona MENU.
La routine richiamata per la lettura delle coordinate dalla tavoletta è la 5.900 che genera sempre un beep, necessario quando si opera visualizzando la pagina HGR2.

Questa routine richiama a sua volta prima la routine 5.700 (che legge dalle loro locazioni i valori assunti in quel momento dalle paddle tramite l'istruzione PEEK) e poi la routine 5.600 che, tramite formule trigonometriche legate alla posizione dei bracci del puntatore, restituisce i valori X%, Y%.

Nella routine 5.900 c'è la necessità di riconoscere letture sbagliate, cosa che avviene quando si punta il menu ed invece il programma aspetta dati grafici.

In tal caso viene richiamata la subroutine 5.400 che segnala la necessità di ricominciare la funzione dall'inizio.

La routine su cui gira tutto il programma è la MAIN (da riga 20 a riga 30).

Qui viene chiamato il codice (gosub 5900), riconosciuta l'area MENU (IF Y% $\leqslant$0) e richiamata la subroutine MENU (riga 6000).

Nella subroutine 6000, tramite una opportuna formuletta matematica, viene individuato il settore Q% puntato e il programma viene inviato alle corrispondenti subroutine.

Lo svolgimento delle operazioni è legato alla logica del programma:

MAIN
MENU
SUBROUTINE (ciascuna con una funzione)
RITORNO AL MAIN

Occorre però fare alcune considerazioni:
Esistono due tipi di funzioni:
- quelle che hanno una lunghezza fissa, per le quali è predeterminato il numero di dati necessari. Ad esempio per tracciare un rettangolo, equidirezionato rispetto agli assi di riferimento, occorrono 5 dati: Codice rettangolo, coordinate X,Y del primo vertice, coordinate X,Y del vertice opposto.
- quelle che hanno una lunghezza variabile, per le quali, non essendo predeterminato il numero dei dati, va comunicato in qualche modo la fine dell'input. Ad esempio la funzione spezzata qualsiasi, può prevedere 2 segmenti come 50.

Per queste funzioni particolari, la condizione di fine spezzata è realizzata puntando la zona menu, che in questo caso non individua una funzione ma solo la fine spezzata.

È questo il sistema più semplice possibile, e comporta come unico inconveniente quello di dover puntare due volte la zona menu per il passaggio da una funzione a lunghezza non predeterminata ad una altra qualsiasi.

Un'altra limitazione è nella possibilità di correzione dati. La correzione può avvenire solo per funzione. Ovvero all'inizio di ogni funzione viene memorizzato il valore del contatore.

Se si commette un errore si richiama la funzione 300, che non fa altro che riportare indietro il contatore.

La correzione avviene però solo a livello dati, non è possibile cancellare la parte sbagliata del disegno.

Diciamo meglio, non è possibile con le funzioni da noi realizzate. È chiaro infatti che, con opportune funzioni, probabilmente un pò più complesse è possibile realizzare una maggiore elasticità e manegevolezza dell'archivio.

Come abbiamo detto le funzioni da noi implementate sono 14. Per le altre 28 abbiamo inserito la istruzione

GOSUB 4.900: RETURN

che visualizza il messaggio:

FUNZIONE Q% NON IMPLEMENTATA.

Quindi se si volesse aggiungere una fun-

```
FUNZIONE 15 PUNTO                DATI N. 1
86   50
FUNZIONE 16 SEGMENTO             DATI N. 4
151  53
128  96
FUNZIONE 17 SPEZZ. QUALS.        DATI N. 9
43   54
79   112
60   126
135  152
148  106
FINE SPEZZATA
FUNZIONE 18 SPEZZ. AP. OR. VERT. DATI N. 20
70   43
100  43
100  73
133  73
FINE SPEZZATA
FUNZIONE 19 SPEZZ. CHI. OR. VERT. DATI N. 26
171  36
201  36
201  65
228  65
228  96
171  96
171  36
FINE SPEZZATA
FUNZIONE 20 RETTANG.             DATI N. 33
97   42
113  83
FUNZIONE 21 CERCHIO              DATI N. 38
137  57
113  106
FUNZIONE 16 SEGMENTO             DATI N. 43
108  82
PUNTO FUORI FORMATO
RIIMMETTERE
107  59
FUNZIONE 1 SWITCH                DATI N. 48
```

Figura 4 - Output alfanumerico su video; mantenendo il programma in posizione TEXT, si può seguire l'andamento dei dati. Ogni tanto si può fare una puntata sulla pagina HGR2.

zione, ad esempio quella richiamata dalla 33.ma casella del menu, basta semplicemente scriverla dalla riga 3.300 fino al massimo alla riga 3.399.

Esaminiamo ora nel dettaglio qualche funzione:

Riga 100 - Funzioni 1 - Switch Text-Hgr2

il flag TH memorizza se si è in pagina testo (TH = 0) o in pagina grafica (TH = 1). A seconda della pagina in cui si è ogni volta che la routine viene chiamata, viene provocato il cambio pagina.

Riga 200 - Funzione 2 - Clear totale

permette il riinizio del programma. Azzera il contatore e passa alla pagina testo.

Riga 300 - Funzione 3 - Correzione Ultima Funzione

all'inizio di ogni funzione che comporta il caricamento nel vettore A% di codici e di dati grafici, viene memorizzato in A0% il valore iniziale del contatore.

In tal modo è possibile, in caso di necessità (ad esempio se si è commesso un errore), riimmettere una funzione

Riga 400 - Funzione 4 - Fine programma.

È l'unica routine di fine, non provoca il caricamento dei dati sul dischetto in quan-

```
10  HGR : HCOLOR= 3: TEXT : GOSUB 5000: GOSUB 5500:I = 1
20  GOSUB 5900: IF Y% < 0 THEN F2 = 1: GOSUB 6000
30 F2 = 0: GOTO 20
100  REM SWITCH TEXT/HGR2
110  IF  NOT TH THEN  POKE 49234,0: POKE 49232,0:TH = 1: GOTO 130
120  POKE 49233,0:TH = 0
130  RETURN
200  REM CLEAR TOTALE
210  HOME : HGR : POKE 49233,0:TH = 0:I = 1: RETURN
300  REM  CORREZIONE ULTIMA FUNCTION
310 I = A0%: RETURN
400  REM  FINE PROGRAMMA
410  IF TH THEN TH = 0
420  GOSUB 100: FOR K = 0 TO 999: NEXT K
430  HOME : TEXT : PRINT "FINE PROGRAMMA ": END
500  GOSUB 4900: RETURN
600  GOSUB 4900: RETURN
700  GOSUB 4900: RETURN
800  GOSUB 4900: RETURN
900  GOSUB 4900: RETURN
1000  GOSUB 4900: RETURN
1100  GOSUB 4900: RETURN
1200  TEXT : HOME : PRINT "SALVATAGGIO SU DISCO PICTURE"
1210  PRINT "------------------------------": PRINT
1220  INPUT "NOME DELLA PICTURE  ";P$:P$ = P$ + ".PIC"
1230  PRINT D$"BSAVE ";P$;",A$2000,L$1FFF"
1240  RETURN
1300  TEXT : HOME : PRINT "SALVATAGGIO SU DISCO FILE"
1305  PRINT "-------------------------": PRINT
1310 I1 = I - 1: FOR H = 1 TO 1999: NEXT : GOSUB 1400
1320  IF TH THEN TH = 0: PRINT
1330  PRINT "PER VISUALIZZARE IL DISEGNO PREMI V"
1335  INPUT "POI PREMI SPACE                    ";V$
1340  IF V$ <  > "V" THEN 1360
1350  GOSUB 100: GET SS$: GOSUB 100
1360  PRINT : PRINT : INPUT "NOME DEL FILE ";NF$
1370  PRINT D$"OPEN ";NF$: PRINT D$"WRITE ";NF$
1380  PRINT I1: FOR J = 1 TO I1: PRINT A%(J): NEXT J
1390  PRINT D$"CLOSE": RETURN
1400  TEXT : HOME
1410  PRINT "STAMPA DATI": PRINT."----------"
1420  PRINT : POKE 34,3: FOR K = 1 TO I - 1
1430 J =  INT (A%(K) - 1000 + 14): PRINT K;
1440  IF A%(K) > 1000 THEN  PRINT  TAB( 13)F$(J): GOTO 1460
1450  PRINT  TAB( 6)A%(K)
1460  NEXT K: FOR J = 1 TO 1999: NEXT : TEXT : RETURN
1500  REM  PUNTO
1510 A0% = I:A%(I) = 1001:F2 = 1
1520  GOSUB 5900:A%(I + 1) = X%:A%(I + 2) = Y%
1530  PRINT X%; TAB( 6)Y%: HPLOT X%,Y%
1540 I = I + 3:F2 = 0: RETURN
1600  REM  SEGMENTO
1610 A0% = I:A%(I) = 1002:F2 = 1
1620  GOSUB 5900:A%(I + 1) = X%:A%(I + 2) = Y%
1630  PRINT X%; TAB( 6)Y%: HPLOT X%,Y%
1640  GOSUB 5900:A%(I + 3) = X%:A%(I + 4) = Y%
1650  PRINT X%; TAB( 6)Y%: HPLOT  TO X%,Y%
1660 I = I + 5:F2 = 0: RETURN
1700  REM   SPEZZATA QUALSIASI
1710 A0% = I:A%(I) = 1003:F2 = 0
1720  GOSUB 5900:A%(I + 1) = X%:A%(I + 2) = Y%
1730  PRINT X%; TAB( 6)Y%: HPLOT X%,Y%:I = I + 3
1750  GOSUB 5900: IF Y% < 0 THEN 1790
1760 A%(I) = X%:A%(I + 1) = Y%
1770  PRINT X%; TAB( 6)Y%: HPLOT  TO X%,Y%
1780 I = I + 2: GOTO 1750
1790  PRINT "FINE SPEZZATA ": RETURN
1800  REM  SPEZZATA ORIZZ.VERT.APERTA
1810 A0% = I:A%(I) = 1004:F2 = 0
1815  GOSUB 5900:A%(I + 1) = X%:A%(I + 2) = Y%
1820  PRINT X%; TAB( 6)Y%: HPLOT X%,Y%:Z% = Y%:I = I + 3
1825  GOSUB 5900: IF Y% < 0 THEN 1880
1830  PRINT X%; TAB( 6)Z%: HPLOT  TO X%,Z%
1835 A%(I) = X%:I = I + 1:Z% = X%
1840  GOSUB 5900: IF Y% < 0 THEN 1880
1845  PRINT Z%; TAB( 6)Y%: HPLOT  TO Z%,Y%
1850 A%(I) = Y%:I = I + 1:Z% = Y%: GOTO 1825
1880  PRINT "FINE SPEZZATA": RETURN
1900  REM  SPEZZATA ORIZZ.VERT.CHIUSA
1910 A0% = I:A%(I) = 1005:F2 = 0
1915  GOSUB 5900:A%(I + 1) = X%:A%(I + 2) = Y%:F% = X%:G% = Y%
1920  PRINT X%; TAB( 6)Y%: HPLOT X%,Y%:Z% = Y%:I = I + 3
1925  GOSUB 5900: IF Y% < 0 THEN 1970
1930  PRINT X%; TAB( 6)Z%: HPLOT  TO X%,Z%
1935 A%(I) = X%:I = I + 1:Z% = X%
1940  GOSUB 5900
1945  PRINT Z%; TAB( 6)Y%: HPLOT  TO Z%,Y%
1950 A%(I) = Y%:I = I + 1:Z% = Y%: GOTO 1925
1970  PRINT F%; TAB( 6)Z%: PRINT F%; TAB( 6)G%
1975  HPLOT  TO F%,Z% TO F%,G%
1980  PRINT : PRINT "FINE SPEZZATA": RETURN
2000  REM  RETTANGOLO QUALSIASI
2010 A0% = I:A%(I) = 1006:F2 = 1
2020  GOSUB 5900:A%(I + 1) = X%:A%(I + 2) = Y%
2030  PRINT X%; TAB( 6);Y%: HPLOT X%,Y%
2040  GOSUB 5900:A%(I + 3) = X%:A%(I + 4) = Y%
2050  PRINT X%; TAB( 6);Y%;
2060  HPLOT  TO X%,A%(I + 2) TO X%,Y%
2070  HPLOT  TO A%(I + 1),Y% TO A%(I + 1),A%(I + 2)
2080 I = I + 5:F2 = 0: RETURN
2100  REM  CERCHIO
2110 A0% = I:A%(I) = 1007:F2 = 1
2115  GOSUB 5900:A%(I + 1) = X%:A%(I + 2) = Y%
2120  PRINT X%; TAB( 6);Y%: HPLOT X%,Y%:CX% = X%:CY% = Y%
2125  GOSUB 5900:A%(I + 3) = X%:A%(I + 4) = Y%
2130  PRINT X%; TAB( 6);Y%: HPLOT X%,Y%
2135 DX = X% - CX% + .00001:DY = Y% - CY%
2140 R =  SQR (DX ^ 2 + DY ^ 2)
2145 AE =  ATN (DY / DX): IF DX < 0 THEN AE = AE + P
2150  FOR J = 0 TO 100:K = J * P / 50
2155 X1% = CX% + R *  COS (K + AE):Y1% = CY% + R *  SIN (K + AE)
2160  IF X1% < 0 OR Y1% < 0 OR X1% > 279 OR Y1% > 191 THEN F3 = 1: GOTO
     2175
2165  IF K = 0 OR F3 THEN  HPLOT X1%,Y1%:F3 = 0: GOTO 2175
2170  HPLOT  TO X1%,Y1%
2175  NEXT J:I = I + 5:F2 = 0: RETURN
2200  GOSUB 4900: RETURN
2300  GOSUB 4900: RETURN
2400  GOSUB 4900: RETURN
2500  GOSUB 4900: RETURN
2600  GOSUB 4900: RETURN
2700  GOSUB 4900: RETURN
2800  GOSUB 4900: RETURN
2900  GOSUB 4900: RETURN
3000  GOSUB 4900: RETURN
3100  GOSUB 4900: RETURN
3200  GOSUB 4900: RETURN
3300  GOSUB 4900: RETURN
3400  GOSUB 4900: RETURN
3500  GOSUB 4900: RETURN
3600  GOSUB 4900: RETURN
3700  GOSUB 4900: RETURN
3800  GOSUB 4900: RETURN
3900  GOSUB 4900: RETURN
4000  GOSUB 4900: RETURN
4100  GOSUB 4900: RETURN
4200  GOSUB 4900: RETURN
4900  REM  FUNZIONE NON IMPLEMENTATA
4910  IF TH THEN  GOSUB 100
4920  PRINT : PRINT "FUNZIONE ";O%;" NON IMPLEMENTATA "
4930  FOR K = 1 TO 999: NEXT : RETURN
5000  REM INIZIALIZZAZIONE
5010  HOME : VTAB (10): FLASH : PRINT " ATTENDERE ": NORMAL
5020  PRINT  CHR$ (4)"BLOAD PADDLE.CODE"
5030  DIM A%(200):P = 3.14159:W = 986:D$ =  CHR$ (4)
5040  DEF  FN PK(I) =  PEEK (I) + 256 *  PEEK (I + 1)
5050 Z0 =  FN PK(797):Z1 =  FN PK(799)
5060 V0 =  FN PK(801):V1 =  FN PK(803)
5070 PY = P / V1:PZ = P / V0
5100  DIM F$(42): FOR K = 1 TO 42: READ F$(K): NEXT K
5110  DATA SWITCH.CLEAR.CORR.ULTIMA FUNCT.,FINE PROGRAMMA,F5,F6,F7,F8,
     F9,F10,F11,SALV.DISCO PICTURE,SALV.DISCO FILE,STAMPA DATI
5120  DATA PUNTO,SEGMENTO,SPEZZ.QUALS.,SPEZZ.AP.OR.VERT.,SPEZZ.CHI.OR.
     VERT.,RETTANG.,CERCHIO,F22,F23,F24,F25,F26,F27,F28
5130  DATA F29,F30,F31,F32,F33,F34,F35,F36,F37,F38,F39,F40,F41,F42
5150  HOME : PRINT "N.  CODICE  'FUNZIONE": PRINT
5160  FOR K = 1 TO 42: PRINT K;: IF K > 14 THEN  PRINT  TAB( 5)W + K;
5170  PRINT  TAB( 11)F$(K): NEXT K: PRINT
5290  RETURN
5400  REM  PUNTO FUORI FORMATO
5410  PRINT "PUNTO FUORI FORMATO                    RIIMMETTERE"
5420  GOSUB 300: RETURN
5500  REM   TARATURA
5510  HOME : VTAB (15): PRINT "     FUNZIONE DI CALIBRAZIONE"
5520  FLASH : PRINT : PRINT " PUNTA IN ALTO A SINISTRA ": NORMAL
5530  GOSUB 5900:X0% = X%:Y0% = Y%
5540  PRINT : PRINT "VALORE X AZZERAMENTO ";X0%
5550  PRINT : PRINT "VALORE Y AZZERAMENTO ";Y0%
5560  FOR K = 1 TO 999: NEXT : HOME : RETURN
5600  REM FORMULE TRIGONOMETRICHE
5610 A = (Z0 - P0) * PZ:B = (P1 - Z1) * PY - A
5620 X% = 150 + ( COS (B) -  COS (A)) - X0%
5630 Y% = 150 + ( SIN (B) +  SIN (A)) - Y0%
5640  RETURN
5700  REM LETTURA VALORI PADDLE
5710  POKE 779,100: CALL 768:P0 = 256 *  PEEK (13) +  PEEK (12)
5720  FOR K = 1 TO 50: NEXT K
5730  POKE 779,101: CALL 768:P1 = 256 *  PEEK (13) +  PEEK (12)
5740  FOR K = 1 TO 50: NEXT K
5750  RETURN
5900  REM  LETTURA
5910  PRINT  CHR$ (7)
5920  GOSUB 5700: IF  PEEK (49251) > 127 THEN 5920
5930  GOSUB 5600
5940  IF Y% < 0 AND F2 THEN  GOSUB 5400: GOTO 5910
5990  RETURN
6000  REM MENU
6010 O% = (3 +  INT (Y% * 3 / 40)) * 14 +  INT (X% / 20) + 1
6020  PRINT : PRINT "FUNZIONE ";O%;" "F$(O%); TAB( 31)"DATI N. ";I
6030  ON O% GOSUB 100,200,300,400,500,600,700,800,900,1000,1100,1200,1
     300,1400,1500,1600,1700,1800,1900,2000,2100,2200,2300,2400,2500,2
     600,2700,2800,2900,3000,3100,3200,3300,3400,3500,3600,3700,3800,3
     900,4000,4100,4200
6040  RETURN
```

Figura 5 - Listato del programma IMMISSIONE; il programma occupa, così com'è, e cioè con poche funzioni implementate, circa 5 K.

to questo compito è espletato dalle due apposite routine.

Righe 500÷1.100

c'è lo spazio per 7 funzioni. In questa zona del menu, ripetiamo, inseriamo routine di utilità, che non provocano cioè il caricamento del vettore.

Riga 1.200 - Funzione 12 - Salvataggio delle Picture

```
FUNZIONE 14 STAMPA DATI
STAMPA DATI
-----------
1               PUNTO
2    86
3    50
4               SEGMENTO
5    151
6    53
7    128
8    96
9               SPEZZ. QUALS.
10   43
11   54
12   79
13   112
14   60
15   126
16   135
17   152
18   148
19   106
20              SPEZZ. AP. OR VERT
21   70
22   43
23   100
24   73
25   133
26              SPEZZ. CHI. OR. VERT.
27   171
28   36
29   201
30   65
31   228
32   96
33              RETTANG.
34   97
35   42
36   113
37   83
38              CERCHIO
39   137
40   57
41   113
42   106
43              SEGMENTO
44   108
45   82
46   107
47   59
```

Figura 6 - Esempio di stampa dati; i dati sono di due tipi. I codici che richiamano una function e i dati X, Y veri e propri.

ci sono due possibilità, quella di conservare il disegno su disco come picture, e quella di conservarlo come archivio dati (funzione 13). Basta indicare il nome da dare alla Picture che questa viene salvata. Il caricamento del file non provoca la fine del programma.

Riga 1.300 - Funzione 13 - Salvataggio di archivi.

Il nostro scopo era quello di creare un archivio di dati grafici codificati. Viene quindi memorizzato il vettore A%.

Come primo dato viene memorizzata la lunghezza del vettore (valore I1) e poi tutti i dati nell'ordine in cui sono stati immessi.

Però prima del caricamento vero e proprio viene richiamata la routine 14 (stampa dati) per controllare i dati "alfanumericamente" e poi viene visualizzato (a richiesta) il disegno realizzato.

Riga 1.400 - Funzione 14 - Stampa dati

La stampa dei dati serve per il controllo, che può essere necessario, dei dati immessi. La stampa, vedi figura 6, riporta il progressivo, l'eventuale codice, con la specifica della funzione relativa, e i dati grafici.

Un discorso, forse un pò banale, va fatto sulle routine di correzione. Chi ha un minimo di dimestichezza con programmi che richiedono un lavoro di input, sa come è facile sbagliare. Se i dati sono tanti poi l'errore è certo.

Occorre quindi che il programma contempli delle routine di correzione che evitino, in caso di errore, di ricominciare da capo. Ma poiché gli errori possono essere di tanti tipi, anche le routine che li correggono debbono essere di tanti tipi. E questo complica il programma, lo appesantisce e in generale provoca altri errori.

Noi abbiamo inserito solo una possibilità di correzione, quella di cancellare l'ultima funzione, dando quindi la possibilità di riimmetterla corretta o di eliminarla.

È peraltro possibile, e lo potrete fare voi, potenziare queste funzioni del programma ad esempio con la routine di correzione di un solo dato o di dati molto "indietro" nell'archivio.

Esaminiamo ora le funzioni che generano dati di archivio.

Riga 1.500 - Funzione 15 - Codice 1001 - Punto

è la funzione più semplice, richiede tre dati: il codice A% (I) = 1001, e le due coordinate A%(I+1) = X%, A%(I+2) = Y%.

Si può notare come la routine, e cosi tutte le altre, svolga funzioni di PLOT e di PRINT.

Il flag F2, settato a 1, predispone il controllo del fuori quadro cui viene sottoposto il dato di input. Tale controllo viene eseguito in riga 5940 solo nel caso che F2 = 1.

Riga 1.600 - Funzione 16 - Codice 1002 - Segmento

è una routine simile a quella del punto. Vengono caricati cinque dati, uno per il codice e gli altri per gli estremi del segmento.

Riga 1.700 - Funzione 17 - Codice 1003 - Spezzata qualsiasi

in questo caso, poiché il numero dei tratti da immettere non è noto a priori, usiamo il fuori quadro come indicazione di fine spezzata (controllo di riga 1750). Il loop che permette il caricamento via via dei segmenti è da riga 1750 a riga 1780.

Riga 1.800 - Funzione 18 - Codice 1004 - Spezzata orizzontale e verticale aperta.

Abbiamo trattato nel numero scorso la routine che permette il risparmio di spazio quando i segmenti della spezzata sono orizzontali o verticali rispetto agli assi di riferimento.

Riportiamo la medesima tecnica anche in fase di input dati.

Anche qui la fine immissione è provocata dal posizionamento del puntatore nella zona menu, e siccome la fine si può verificare sia dopo una linea orizzontale che dopo una verticale, la condizione va testata due volte (riga 1825 e riga 1840).

Riga 1.900 - Funzione 18 - Codice 1005 - Spezzata orizzontale e verticale aperta

analoga alla precedente solo che la linea è chiusa. Per chiudere la spezzata occorre memorizzare il punto di inizio. La condizione di fine può verificarsi solo dopo l'immissione di un segmento verticale in quanto la "chiusura" della linea è obbligata.

Ad esempio se debbo immettere un rettangolo, immetto il primo punto (X1, Y1), poi il secondo (X2, Y1) che sta sulla retta orizzontale che passa per i due punti, poi il terzo sulla verticale (X2, Y2). Se ora comunico la fine spezzata individuo automaticamente il penultimo punto (X1, Y2) e l'ultimo (X1, Y1) che altri non è che il primo.

Occorrono quindi solo 5 dati.

Riga 2.000 - Funzione 20 - Codice 1006 - Rettangolo qualsiasi

Occorrono 5 dati, il codice 1006 e le coordinate dei due vertici di una diagonale.

Riga 2.100 - Funzione 21 - Codice 1007 - Cerchio

La funzione vuole il centro ed un punto qualsiasi.

Nel software della tavoletta ci sono anche funzioni ARC e CERCHIO dati tre punti. Nel nostro caso abbiamo inserito solo il cerchio dati due punti.

Inoltre nel software di Bo Arnklit viene memorizzata una tabella di funzioni trigonometriche. Noi non lo facciamo per poter economizzare spazio anche se il disegno del cerchio diviene molto lento.

La funzione contiene il controllo del fuori quadro, e questo rallenta ulteriormente l'esecuzione.

Il controllo se il punto risulta fuori quadro interrompe il disegno (con Hplot to di riga 2170) che viene ripreso quando il punto risulta interno al quadro (con Hplotto di riga 2165).

Il programma poteva essere compattato ulteriormente, raccogliendo in subroutine le istruzioni più volte ripetute nelle varie funzioni di caricamento dati.

Ad esempio si vedano le coppie di righe 1520 - 1530, 1620 - 1630, 1720 - 1730, ecc. che sono identiche.

Infine abbiamo escluso funzioni che richiedono il caricamento di codici in linguaggio macchina per evitare problemi di contabilità con le pagine grafiche.

Sul prossimo numero, continuando l'argomento, suggeriremo soluzioni a questo problema. MC