

Archivi per dati di grafica edile ed architettonica con la tavoletta grafica di MCmicrocomputer

Torniamo alla grafica bidimensionale per trattare la gestione di archivi grafici per applicazioni particolari.

Vogliamo costruire, con l'aiuto dell'Apple II e della tavoletta di MC, un archivio grafico per dati architettonici, che poi visualizzeremo su un video o su plotter.

Chi usa la tavoletta grafica di MC ne ha sicuramente apprezzato le potenti funzioni grafiche che permettono di realizzare qualsiasi immagine su video e permettono la sua memorizzazione come PICTURE FILE su dischetto. Noi vogliamo potenziare tale software indirizzandolo a funzioni di archivio vero e proprio e non a semplice funzione di salvataggio di singole immagini.

Ambientiamoci guardando la figura 1. I nostri strumenti sono un Apple II che ha uscita su video 280 per 192 pixel, una tavoletta grafica di MC che ha la possibilità di essere collegata direttamente in uscita sul video Apple II ed ha quindi una definizione di 280 per 192 millimetri.

In realtà la tavoletta permette, come abbiamo visto in articoli precedenti, una maggiore definizione che in taluni casi conviene sfruttare.

In uscita utilizzeremo il video grafico Apple II o il plotter Watanabe.

L'applicazione che presentiamo in questo numero prevede il disegno di una piantina di una casa arredata. La scala del disegno è 1:40 ovvero 1 mm = 4 cm, ovvero sul video 1 pixel = 4 cm. È una applicazione che può essere facilmente implementata con ulteriori subroutine e con ulteriori funzioni: ad esempio con funzioni di cancellazione, di movimento di elementi, ecc. Si può ipotizzare di eseguire una progettazione edile direttamente con il computer.

Oppure si possono creare funzioni opportune per il disegno di particolari architettonici o di arredamento o di impianti. Si pensi al disegno in pianta di un camino, o di un divano a tre posti o di un impianto elettrico.

La trattazione dell'argomento è suddivisa in due parti:

- la fase di input e quindi di codifica dei dati
- la fase di output e quindi di decodifica dei dati.

Tratteremo dapprima il secondo argomento anche se questo può sembrare illogico in quanto è vedendo i risultati che capiremo meglio i problemi connessi con la codifica dei dati grafici.

Questo è il tema dell'articolo di questo numero. Sul prossimo presenteremo un programma, basato sull'uso della tavoletta grafica, che costruisce i dati secondo la stessa codifica che vedremo nei programmi di output.

In sostanza i programmi presentati oggi

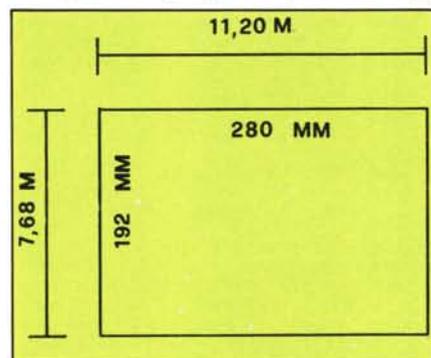


Figura 1 - Ambiente Digitizer/Video - In scala 1:40 il formato permesso dal piano della tavoletta è 11,20 m per 7,68 m. Sul video la scala è 1 pixel = 4 cm.

```

100 REM TRADIZIONALE
110 DIM X%(7), Y%(7): FOR I = 1 TO 7: READ X%(I), Y%(I): NEXT
120 HGR2: HCOLOR= 3: FOR I = 1 TO 6
130 HPLT X%(I), Y%(I) TO X%(I + 1), Y%(I + 1): NEXT
140 REM SHORT
150 DIM A%(9): FOR I = 1 TO 9: READ A%(I): NEXT: I = 1
160 IF A%(I) = 1001 THEN I = I + 1: GOSUB 170: END
170 REM SUB SPEZZATA CHIUSA ORIZZ VERT.
180 HPLT A%(I), A%(I + 1): B1% = A%(I): B2% = A%(I + 1)
190 HPLT TO A%(I + 2), A%(I + 1): REM ORIZ
200 I = I + 3: IF A%(I) > 1000 THEN HPLT TO B1%, B2%: RETURN
210 HPLT TO A%(I - 1), A%(I): REM VERT.
220 I = I - 1: GOTO 190
230 DATA 20, 140, 130, 140, 130, 100, 80, 100, 80, 20, 20, 20, 140
240 DATA 1001, 150, 140, 260, 100, 200, 20, 150, 1200
    
```

Figura 2 - Listato del programma Short - Viene eseguito il disegno del poligono di figura 3 con i due metodi, quello tradizionale e quello abbreviato, valido per spezzate di segmenti orizzontali e verticali.

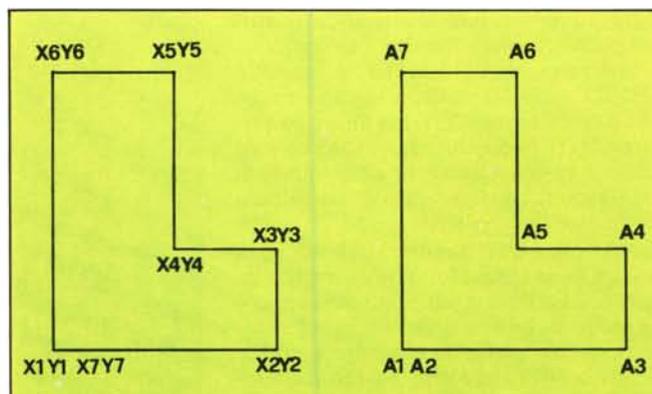


Figura 3 - Disegno del poligono realizzato con il programma Short. Si tratta di trovare il metodo più economico per disegnare il poligono formato da sei vertici e da sei segmenti.

sono già utilizzabili solo che i dati vanno immessi "a mano" e inseriti "a mano" nei DATA del programma. Il che non è molto gravoso se si utilizzano opportuni accorgimenti come, ad esempio, quello di riportare il disegno su carta millimetrata.

Si pensi che il disegno della figura 9 è costruito con circa 350 dati elementari opportunamente codificati.

Abbiamo realizzato alcuni programmi che isolano ed approfondiscono funzioni implementate nel programma generalizzato DISEGNO, soprattutto poiché possono costituire a loro volta moduli per i vostri programmi. Abbiamo isolato le funzioni di disegno di linea spezzata chiusa, formata da segmenti orizzontali e verticali, la funzione di Fill, la funzione di Text.

Altre funzioni tipo Box, Circle, ecc. sono facilmente estraibili dal programma DISEGNO. Altre, infine, sono inseribili altrettanto facilmente, in quanto per eseguirle basta identificarle con un codice numerico.

Programma SHORT

Nei disegni architettonici spesso le figure sono composte da linee orizzontali e verticali in sequenza. Questo permette un cospicuo risparmio di dati: infatti, per definire un segmento occorrono quattro coordinate, ma se il segmento è orizzontale ne occorrono solo tre, in quanto la coordinata Y dei due punti, che individuano il segmento, è la stessa.

Se i segmenti sono due in sequenza, ne occorrono sei, ma se il primo è orizzontale ed il secondo verticale ne occorrono solo quattro e così via. Cioè per una spezzata qualsiasi costituita da N segmenti servono $2 * (N + 1)$ coordinate.

Se la sequenza è di linee orizzontali e verticali occorrono solo $N + 2$ coordinate. Riferendoci ad una linea chiusa (fig. 3) vedremo che ragionando in termini di cop-

```

100 PRINT CHR$(4)"BLOADASS.CODE"
110 HGR
120 HOME : VTAB (21): INPUT "X1, Y1 " : X1, Y1
130 HCOLOR= 3: INPUT "X2, Y2 " : X2, Y2
140 X% = (X1 + X2) / 2: Y% = (Y1 + Y2) / 2
150 PRINT : INPUT "N. COLORE " : AC
160 HPLLOT X1, Y1 TO X1, Y2 TO X2, Y2 TO X2, Y1 TO X1, Y1
170 GOSUB 180: GOTO 120
180 REM FILL
190 POKE 6448, AC / 32: POKE 6423, AC - INT (AC / 32) * 32
200 POKE 6418, X% / 256: POKE 6417, X% - INT (X% / 256) * 256
210 POKE 6416, Y%: CALL 6449: RETURN

```

Figura 4 - Listato del programma Fill - Il programma permette il riempimento (con un colore scelto tra 128) di un rettangolo di cui siano dati i due vertici.

```

100 REM
110 REM SPEZZATA
120 HGR2 : HCOLOR= 3
130 X% = RND (1) * 280: Y% = RND (1) * 192
140 HPLLOT X%, Y%: FOR I = 1 TO 50
150 X% = RND (1) * 280: HPLLOT TO X%, Y%
160 Y% = RND (1) * 192: HPLLOT TO X%, Y%: NEXT I
190 REM
200 REM FILL
210 FOR I = 1 TO 50: C% = RND (1) * 128
220 X% = RND (1) * 280: Y% = RND (1) * 192
230 GOSUB 300: NEXT I: END
290 REM
300 REM SUB-FILL
310 POKE 6448, C% / 32: POKE 6423, C% - INT (C% / 32) * 32
320 POKE 6418, X% / 256: POKE 6417, X% - INT (X% / 256) * 256
330 POKE 6416, Y%: CALL 6449: RETURN

```

Figura 5 - Listato del programma Casual - È una specie di "merge" tra i due programmi precedenti. La linea spezzata è del tutto casuale, così come i riempimenti di colore.

pie di coordinate ne occorrono 7, per un totale di 14 elementi. Invece codificando la sequenza ne occorrono solo 7, in quanto anche l'ultimo punto, quello di chiusura, può essere omesso se c'è un codice di chiusura della spezzata.

Dalla tabella di comparazione si vede quali dati non sono necessari (sono quelli in neretto):

X1 Y1 X2 Y2 X3 Y3 X4 Y4 X5 Y5 X6 Y6
X7 Y7
A1 A2 A3 A2 A3 A4 A5 A4 A5 A6 A7 A6
A1 A2

Abbiamo realizzato il programma SHORT che evidenzia la differenza tra i due metodi.

Il caricamento dei DATA nel secondo caso è agevole in quanto basta immettere il punto di partenza (A1, A2), la coordinata della X del secondo punto (A3), (N.B. va scelto nel nostro caso come primo segmento un segmento orizzontale) e poi le coordinate X, Y che cambiano.

Il programma interpreta valori A(I) > 1000 come codici che specificano cosa si deve fare. Ogni sequenza di dati grafici è preceduta da un codice A (I) > 1000 e ne è seguita.

Esaminiamo il listato di figura 2

Da riga 110 a riga 130 c'è il disegno della spezzata chiusa ottenuto con il metodo delle coppie di coordinate. Un poligono come quello di figura 3 è individuato da 6 segmenti e da 6 coppie di punti e per poter utilizzare un unico loop di plottaggio dei segmenti occorre individuare anche il settimo punto (coincidente con il primo).

Dalla riga 140 alla riga 160 c'è il metodo SHORT, che innanzi tutto non utilizza coordinate cartesiane X, Y ma un unico vettore A%(I). In questo vettore sono caricati i codici e, conseguentemente, i valori di grandezza il cui significato è individuato dal codice che li precede.

Nel nostro caso il codice 1001 individua una spezzata chiusa formata da tratti orizzontali e verticali il cui primo segmento è orizzontale. La fine della spezzata è automaticamente definita dal nuovo codice A%(I) > 1000 che viene trovato nella sequenza. La subroutine 170 è generalizzata, ovvero è utilizzabile con altri dati in un altro programma. La riga 180 traccia

```

100 REM INIZIALIZZAZIONI
110 D$ = CHR$(4): ONERR GOTO 160
120 PRINT D$"BLOADASS.CODE": PRINT D$"BLOADASCII.SET, A#8000"
130 POKE 232, 0: POKE 233, 128
140 HOME : DIM A%(500): FOR I = 1 TO 500: READ A%(I): NEXT
150 REM MAIN PROGRAM
160 I = 1: HGR2 : HCOLOR= 3
170 I = I + 1: ON A%(I - 1) - 1000 GOSUB 1010, 1110, 1210, 1310, 1410, 1510, 1
610, 1710, 1810, 1910, 2010
180 IF A%(I) > 1100 THEN FOR K = 1 TO 3000: NEXT K: HOME : END
190 GOTO 170
1000 REM SPEZZATA CHIUSA
1010 HPLLOT A%(I), A%(I + 1): B1% = A%(I): B2% = A%(I + 1)
1020 HPLLOT TO A%(I + 2), A%(I + 1)
1030 I = I + 3: IF A%(I) > 1000 THEN HPLLOT TO B1%, B2%: RETURN
1040 HPLLOT TO A%(I - 1), A%(I)
1050 I = I - 1: GOTO 1020
1100 REM FINESTRA ORIZZ
1110 HPLLOT A%(I), A%(I + 1) TO A%(I + 2), A%(I + 1): HPLLOT A%(I), A%(I + 3
) - 2 TO A%(I + 2), A%(I + 3) - 2
1120 HPLLOT A%(I), A%(I + 3) + 2 TO A%(I + 2), A%(I + 3) + 2: I = I + 4: RETURN
1200 REM FINESTRA VERT
1210 HPLLOT A%(I), A%(I + 1) TO A%(I), A%(I + 3): HPLLOT A%(I + 2) - 2, A%(I
+ 1) TO A%(I + 2) - 2, A%(I + 3)
1220 HPLLOT A%(I + 2) + 2, A%(I + 1) TO A%(I + 2) + 2, A%(I + 3): I = I + 4
: RETURN
1300 REM PORTA IN SU
1310 D% = ABS (A%(I + 2) - A%(I)): HPLLOT A%(I), A%(I + 1) TO A%(I + 2), A
%(I + 1) TO A%(I), A%(I + 1) + D% TO A%(I), A%(I + 1): I = I + 3: RETURN
1400 REM PORTA IN GIU
1410 D% = ABS (A%(I + 2) - A%(I)): HPLLOT A%(I), A%(I + 1) TO A%(I + 2), A
%(I + 1) TO A%(I), A%(I + 1) - D% TO A%(I), A%(I + 1): I = I + 3: RETURN
1500 REM SPEZZATA APERTA
1510 HPLLOT A%(I + 1), A%(I): I = I + 1
1520 HPLLOT TO A%(I), A%(I + 1): I = I + 1: IF A%(I + 1) > 1000 THEN I =
I + 1: RETURN
1530 HPLLOT TO A%(I + 1), A%(I): I = I + 1: IF A%(I + 1) > 1000 THEN I =
I + 1: RETURN
1540 GOTO 1520
1600 REM ARMADIO
1610 HPLLOT A%(I), A%(I + 1) TO A%(I), A%(I + 3) TO A%(I + 2), A%(I + 3) TO
A%(I + 2), A%(I + 1) TO A%(I), A%(I + 1) TO A%(I + 2), A%(I + 3): I = I
+ 4: RETURN
1700 REM LETTO
1710 HPLLOT A%(I), A%(I + 1) TO A%(I), A%(I + 3) TO A%(I + 2), A%(I + 3) TO
A%(I + 2), A%(I + 1) TO A%(I), A%(I + 1) TO A%(I) + 10, A%(I + 3): I =
I + 4: RETURN
1800 REM RETTANGOLO GENERICO
1810 HPLLOT A%(I), A%(I + 1) TO A%(I), A%(I + 3) TO A%(I + 2), A%(I + 3) TO
A%(I + 2), A%(I + 1) TO A%(I), A%(I + 1): I = I + 4: RETURN
1900 REM FILL
1910 AC = A%(I)
1920 X% = A%(I + 1): Y% = A%(I + 2)
1930 POKE 6448, AC / 32: POKE 6423, AC - INT (AC / 32) * 32
1940 POKE 6418, X% / 256: POKE 6417, X% - INT (X% / 256) * 256: POKE 641
6, Y%
1950 CALL 6449
1960 I = I + 2: IF A%(I + 1) > 1000 THEN I = I + 1: RETURN
1970 GOTO 1920
2000 REM TEXT
2010 RT = A%(I): SC = A%(I + 1): ROT= RT: SCALE= SC: HCOLOR= 3
2020 I = I + 2: CS = COS (RT * P / 32): SN = SIN (RT * P / 32)
2030 X% = A%(I): Y% = A%(I + 1): CR = A%(I + 2): I = I + 2
2040 IF CR > 1000 THEN RETURN
2050 DRAW CR AT X%, Y%
2060 X% = X% + 8 * SC * CS: Y% = Y% + 8 * SC * SN
2070 I = I + 1: CR = A%(I): GOTO 2040

```

```

100 PRINT CHR# (4)"BLOADASCII.SET.A#8000"
110 POKE 232,0: POKE 233,128
120 HOME : ONERR GOTO 150
130 DIM A%(100):P = 3.1416: HGR2 : HCOLOR= 3:I = 1
140 READ A%(I):I = I + 1: GOTO 140
150 REM MAIN
160 I = 1
170 IF A%(I) > 1100 THEN END
180 I = I + 1: GOSUB 200: GOTO 170
200 REM SUB-SCRIPT
210 RT = A%(I): SC = A%(I + 1): SCALE= SC: ROT= RT:I = I + 2
220 CS = COS (RT * P / 32):SN = SIN (RT * P / 32)
230 X% = A%(I):Y% = A%(I + 1):CR = A%(I + 2):I = I + 2
240 IF CR > 1000 THEN RETURN
250 IF CR > 1000 THEN RETURN
260 DRAW CR AT X%,Y%
270 X% = X% + 8 * SC + CS
280 Y% = Y% + 8 * SC + SN
290 I = I + 1:CR = A%(I): GOTO 240
999 REM DATA
1000 DATA 1010,0,1,100,100,66,67,68,69,70,71,72,73,74,75,76,77
1010 DATA 1010,16,1,66,78,79,80,81,82,83,84
1020 DATA 1010,32,2,200,150,66,85,86,87,88,89,90
1030 DATA 1010,48,1,100,100,66,91,90,89,88,87,86,85
1040 DATA 1200

```

Figura 7 - Listato del programma Text. Anche qui è tutta questione di codifica. Per ogni parola occorre specificare angolo di rotazione, scala e coordinate iniziali.

```

10000 DATA 1001,5,5,40,15,15,40,5
10010 DATA 1001,5,70,15,110,45,115,15,130,5
10020 DATA 1001,5,160,15,179,90,189,5
10030 DATA 1001,120,189,160,179,140,120,157,115,140,110,130,100,120
10040 DATA 1001,190,189,275,170,265,179,190
10050 DATA 1001,265,140,275,95,265,115,195,100,190,115,180,120,265
10060 DATA 1001,265,65,275,5,240,15,265
10070 DATA 1001,210,15,195,77,189,75,179,70,189,15,175,5,210
10080 DATA 1001,120,5,155,15,140,70,157,75,140,80,130,15,120
10090 DATA 1001,70,5,90,15,70
10100 DATA 1002,40,15,70,5
10110 DATA 1002,155,15,175,5
10120 DATA 1002,210,15,240,5
10130 DATA 1002,90,179,120,189
10140 DATA 1002,160,179,190,189
10150 DATA 1003,15,70,5,40
10160 DATA 1003,15,130,5,160
10170 DATA 1003,265,65,275,95
10180 DATA 1003,265,140,275,170
10190 DATA 1004,157,120,180
10200 DATA 1004,90,10,105
10210 DATA 1004,120,10,105
10220 DATA 1005,157,70,180
10230 DATA 1005,195,100,218
10240 DATA 1006,5,80,0,130,5
10250 DATA 1006,179,85,130,65,164,25,130,45,115
10260 DATA 1007,141,121,155,178
10270 DATA 1007,196,114,264,100
10280 DATA 1008,196,31,245,49
10290 DATA 1008,196,51,245,69
10300 DATA 1008,213,121,263,139
10310 DATA 1008,191,159,240,177
10320 DATA 1009,241,177,264,165
10330 DATA 1009,180,121,212,133
10340 DATA 1009,141,69,156,35
10350 DATA 1009,144,66,153,38
10360 DATA 1009,142,20,154,30
10370 DATA 1009,176,20,188,30
10380 DATA 1009,179,40,189,58
10390 DATA 1009,181,42,186,56
10400 DATA 1010,1,10,10,80,10,10,100,132,30,132,135,192,30,273,30,273,
118,273,185,192,110
10410 DATA 1010,50,160,65,170,20,170,34,185,65
10420 DATA 1011,0,1,34,70,84,66,77,80,79,70
10430 DATA 1011,0,1,24,150,68,86,68,74,79,66
10440 DATA 1011,0,1,200,145,77,70,85,85,80
10450 DATA 1011,0,1,220,85,77,70,85,85,80
10460 DATA 1011,0,1,150,90,69,74,84
10470 DATA 1011,0,1,160,40,67,66
10480 DATA 1200

```

Figura 8 - Listato del programma Disegno - Il MAIN PROGRAM è elementare. Può essere implementato aggiungendo ulteriori subroutine, identificate dai codici 1012, 1013, 1014, ecc.

il primo punto di coordinate $A\%(I)$, $A\%(I+1)$ di cui conserva i valori necessari per la chiusura. La riga 190 traccia la generica linea orizzontale, si vede che il valore della coordinata Y è uguale a quello del punto precedente. La riga verticale è tracciata dalla riga 210.

La condizione di fine spezzata si verifica quando $A\%(I) > 1000$ e produce la chiusura e il ritorno dalla subroutine. Non ci sono grosse difficoltà logiche nella utilizzazione di questo tipo di codifica. L'unica cosa un po' delicata è il contatore I che deve sempre individuare correttamente gli elementi e che quindi va calcolato con attenzione. È facile che in fase di codifica cerchiate di plottare un numero codice ($A\%(I) > 1000$).

Filler

Un'altra funzione da implementare è quella di fill, che come noto permette di riempire con un colore a scelta una qualsiasi figura chiusa.

La funzione fill è stata presa "di peso" dal software della tavoletta di MC. Il listato è in figura 4.

Per utilizzare il fill occorrono le seguenti informazioni: CODICE, N. COLORE, COORDINATE del punto interno alla linea chiusa.

Il programma dimostrativo carica l'"ASS.CODE", che contiene le routine in linguaggio macchina necessarie alla operazione (riga 100). Poi, accesa la pagina HGR, vengono chieste le due coordinate che definiscono un Box (righe 120-130) e vengono calcolate le coordinate $X\%$, $Y\%$ di un punto sicuramente interno al box (riga 150) e infine viene chiesto il colore. Il programma plotta la scatola e poi esegue la routine di riga 180 che "riempie" di colore AC il box. A tale routine sono comunicati i valori AC, $X\%$, $Y\%$ che vengono immessi nelle opportune locazioni di memoria.

Casual

Dei due programmi precedenti è stato fatto un "merge" (listato in fig. 5). Viene tracciata, con la routine 110-160, una spezzata random costituita da 50 segmenti variamente intersecantisi. Poi in questa maglia irregolare vengono individuati a caso dei punti e vengono riempiti di colore a caso i poligonetti individuati (righe 200-230). La routine di riempimento vero e proprio è in riga 300 ed è identica a quella provata nel programma precedente.

Il risultato è presentato in figura 6.

Text

Un'altra funzione fondamentale è quella che permette di scrivere parole e frasi, con un set di caratteri a scelta, sulla pagina grafica su cui stiamo disegnando.

Anche questa funzione l'abbiamo prelevata dal software della tavoletta, modificandola opportunamente. Il programma che ne abbiamo tratto è in figura 7.

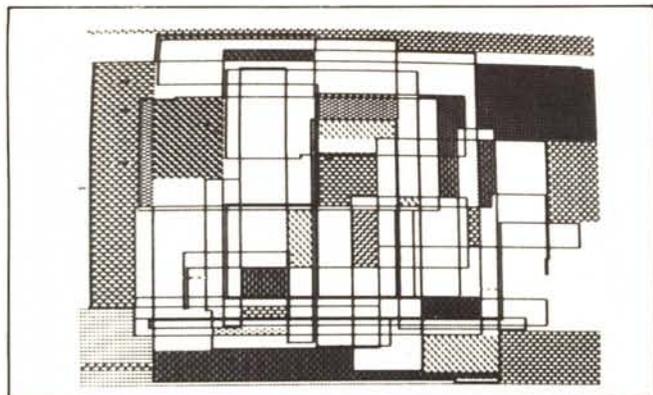


Figura 6 - Output del programma Casual - Segmenti, colori e aree da riempire sono casuali.

Dapprima carichiamo il set di caratteri che vogliamo utilizzare. I set di caratteri sono quelli forniti con il software della tavoletta grafica. A loro volta questi set sono stati tradotti in formato shape-table dal formato Hires Character Table che avevano nel software del Tool Kit dal quale sono stati prelevati.

Nel software della tavoletta il set di caratteri è posizionato nella locazione di memoria destinata alla pagina HGR2. Nel nostro caso, poiché vogliamo lavorare proprio sulla pagina HGR2, lo carichiamo un po' più su. (Riga 100).

I DATA sono così codificati:

1010 individua che stiamo lanciando un messaggio di TEXT.

0,1 i due valori successivi rappresentano i valori ROT, SCALE della shape-table; 100,100 indicano le coordinate di partenza della frase;

65,66 i valori successivi rappresentano i valori, secondo l'ordine delle shape, dei vari caratteri.

La funzione di READ dei dati è gestita dalla funzione di ERROR. Poiché il programma non conosce il numero dei DATA, legge ad oltranza fino ad un END OF DATA ERROR, che significa che i DATA sono finiti. (Righe 110-140). È opportuno ovviamente inserire la funzione ERROR solo quando il programma è sicuramente perfetto altrimenti anche altri tipi di errore vengono interpretati come END OF DATA. D'altra parte, far gestire dalla routine di ERROR il caricamento dei dati consente di immetterli via via senza doverli contare. A questo punto il contatore viene settato a 1 e comincia il MAIN PROGRAM. Il contatore viene via via incrementato di una unità fin quando non viene trovato il codice di fine codifica $A\% (I) > 1.000$.

La subroutine di text (da riga 200 a riga 290) individua subito RT,SC valori di rotazione e scala che non variano per tutta la riga di stampa. Individua poi i valori $X\%$, $Y\%$, posizione del primo carattere (righe 210-230). Le posizioni dei caratteri successivi sono calcolate in funzione di SC, fattore moltiplicativo e CS,SN funzioni trigonometriche legate alla rotazione.

A tal punto viene individuato il CR nu-

mero progressivo del carattere da disegnare nel set a disposizione. (Riga 290).

SET SCALE e ROT non cambiano, $X\%$, $Y\%$ sono calcolati automaticamente, i singoli caratteri sono codificati uno dopo l'altro e tracciati nel ciclo tra le righe 240-290.

Si va avanti fin quando non si trova un codice $A\% (I) > 1.000$ che provoca il ritorno dalla subroutine.

In totale per scrivere una frase di N caratteri in un punto qualsiasi dello schermo occorrono $N - 5$ dati.

Disegno

Passiamo ora al programma generale. Il listato è in figura 8 ed è diviso in due pezzi, il primo contiene il programma vero e proprio, il secondo contiene i DATA che producono esattamente la figura 9.

I DATA sono immessi in modo che ad ogni riga corrisponde un codice con i dati relativi alla funzione corrispondente. Ad esempio i DATA di riga 10.260 iniziano con un codice 1.007, che equivale al disegno di un armadio. Un armadio si disegna con un rettangolo e una diagonale. Quindi occorrono le quattro coordinate dei due vertici del rettangolo e la diagonale si traccia tra questi stessi due vertici.

Va notato come alcune funzioni necessitano di un numero fisso di elementi, ad esempio per disegnare un armadio ne occorrono quattro, altre funzioni possono avere un numero variabile di dati. In questo caso, o meglio in queste subroutine, saranno inseriti dei test di riconoscimento fine routine coincidenti con un nuovo codice. La struttura del programma è semplicissima. C'è una parte di lettura codici (righe 100-140). Una parte MAIN di individuazione codici e rimando alle subroutine (righe 150-190). Una parte contenente le varie subroutine (ne abbiamo implementate 11), e infine c'è la parte con i DATA.

C'è da notare che possono sorgere dei problemi per programmi molto lunghi quando si vogliono inserire contemporaneamente più routine in linguaggio macchina per l'esecuzione delle varie funzioni. Nel nostro caso le funzioni di Fill e di Text.

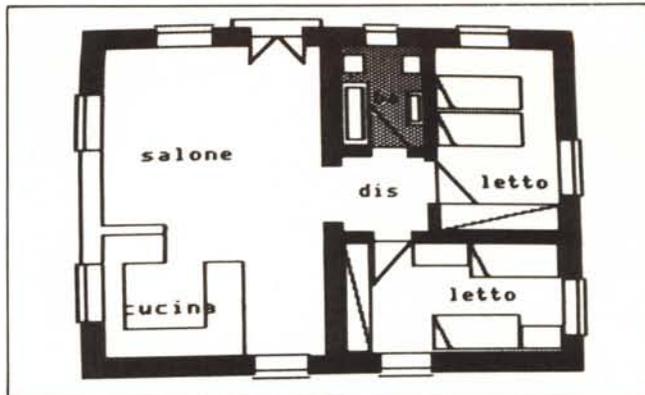


Figura 9 - Output del programma Disegno - Con il sistema di codifica descritto, per realizzare questo disegno sono necessari 345 dati, tra codici, dati di specificazione e coordinate.

Infatti poiché si disegna sulla pagina HGR2, se il programma è molto lungo, ovvero ha molte funzioni o molti dati, possono sorgere problemi di spazio. In questo caso si può facilmente spezzare il programma in due eseguendo, ad esempio, con un programma tutto il disegno, e con un secondo programma, richiamato alla fine del primo e che carica le routine in linguaggio macchina, la funzioni Fill e Text.

Le funzioni implementate sono:

SPEZZATA DI LINEE (cod. 1001): già vista nel programma Short.

FINESTRA ORIZZONTALE (cod. 1002) e FINESTRA VERTICALE (cod. 1003) per le quali occorrono quattro dati. La funzione finestra produce il disegno di tre segmenti paralleli.

PORTA IN SU (cod. 1004) e PORTA IN GIÙ (cod. 1005), necessitano di quattro elementi e producono il disegno dei triangolini appunto con la punta in su o in giù.

SPEZZATA APERTA (cod. 1006) è identica alla spezzata chiusa, solo che non ha la funzione di chiusura. Ha il test di fine dopo il tracciamento di ciascun segmento.

FILLER (cod. 1007) già visto nell'apposito programma DEMO, richiede numero di colore, e poi via via coppie di coordinate.

ARMADIO (cod. 1007), LETTO (cod. 1008), RETTANGOLO GENERIC (cod. 1009) richiedono tutti quattro coordinate.

FILLER (cod. 1010) già visto nell'apposito programma DEMO, richiede numero di colore, e poi via via coppie di coordinate. Per cambiare colore occorre un nuovo codice, che produce un nuovo ingresso alla subroutine.

TEXT (cod. 1011) anche questa subroutine l'abbiamo già descritta più volte con l'apposito programma omonimo, e quindi non c'è altro da dire.

Suggeriamo a chi volesse provare il programma di caricare a pezzetti sia le routine, sia i DATA in modo da provare via via le funzioni e di capirne il meccanismo. E soprattutto, per i problemi di spazio citati, suggeriamo, di escludere i REM.

Comprereste un'automobile sconosciuta?

Una grande industria significa grande produzione. E per vendere con successo grandi quantitativi, il prodotto deve essere altamente sviluppato ed essere attrattivo per la vendita per un lungo periodo di tempo. Non c'è spazio per i compromessi. Il prodotto deve soddisfare le necessità del mercato, nelle sue varie componenti. Ma questo già lo sapevate.

TOSHIBA è nel mondo una delle Aziende Leader nel campo dell'elettronica, con più di 100.000 dipendenti.

TOSHIBA ha una incomparabile esperienza tecnica.

TOSHIBA costruisce computers da più di 20 anni e li vende con grande successo nel mercato Giapponese dove solo i migliori sopravvivono.

Ora i computers TOSHIBA sono disponibili anche in Italia:

il T 100 ne è un esempio.

È uno dei più versatili microcomputer in commercio, con un "magazzino" memoria ampliabile studiato per soddisfare anche le Vs. necessità future. Ha il collegamento diretto con video verde e/o a colori, video a cristalli liquidi, televisione, floppy-disk drive, audio cassetta e stampante. Voi non potete permettere di lasciarVi sfuggire l'occasione di utilizzare il T 100.

Technical data

CPU	Z - 80A (4MHz), Interfaccia RS - 23C, IEEE-488
Memoria	ROM 32-64 KB, RAM 64-96 KB
Video	RAM 16 KB
Display	25 linee, 80 caratteri, 640 x 200 dots
Floppies	5 1/4", 280 KB
Stampante	80 or 136 cm, 120 cps
Sistema Operativo	CP/M, PASCAL, T-BASIC

Personal computer T 100



Informiamo i Sigg.ri Agenti e Rivenditori Software-houses che abbiamo ancora alcune zone libere. Chi è interessato può contattarci per ulteriori informazioni.

TIBER ATTREZZATURE UFFICIO SPA
Via Madonna del Riposo, 127
00165/ROMA

Vogliate inviarmi a giro di posta
ulteriori notizie del TOSHIBA T 100

TOSHIBA

Name _____



A CIASCUNO IL SUO

HP85 PERSONAL COMPUTER INTEGRATO PORTATILE
PER APPLICAZIONI TECNICO SCIENTIFICHE

HP86 COMPUTER MODULARE AD ALTE PRESTAZIONI
PER APPL. SCIENTIFICHE E GESTIONALI

HP9816 PERSONAL COMPUTER A 16/32 BIT PER APPL.
AD ALTA VELOCITA' DI ELABORAZIONE

HP120/125 COMPUTER PER UFFICIO PER APPLICAZIONI
MANAGERIALI E DI WORD PROCESSING



Ogni applicazione richiede prestazioni diverse. L'esperienza SILVERSTAR vi aiuta a scegliere la soluzione più adatta alle vostre esigenze, abbinando l'alta qualità dei computer HP ad una serie completa di programmi applicativi.

	HP 85	HP 86	HP 125	HP 9816
Mem. RAM fino a	32 K	576 K	64 K	768 K
Sistema operat./Linguaggi	Basic HP	Basic HP CPM/Pascal	CPM	Basic/Pascal HPL
Video	alfanumer./grafico	alfanum./grafico (esterno)	alfanumerico	alfanumer./grafico
Interfacce interne	—	Parall. Centronics	HP IB+2 RS232C	HP IB, RS232
Periferiche interne	Printer e cart. magnetica	—	—	—

Se siete interessati ai personal computer HP compilate e spediteci questo tagliando.

Cognome

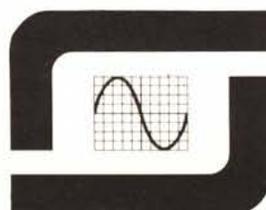
Nome

Qualifica Azienda

Via

CAP Città

Telefono



silverstar
componenti e sistemi

Sede: 20146 Milano - Via dei Gracchi, 20 - Tel. (02) 4996 (12 linee) - Telex 332189
40122 Bologna - Via del Porto, 30 - Tel. (051) 522231
00198 Roma - Via Paisiello, 30 - Tel. (06) 8448841 (5 linee) - Telex 610511
10139 Torino - P.za Adriano, 9 - Tel. (011) 443275/6 - 442321 - Telex 220181