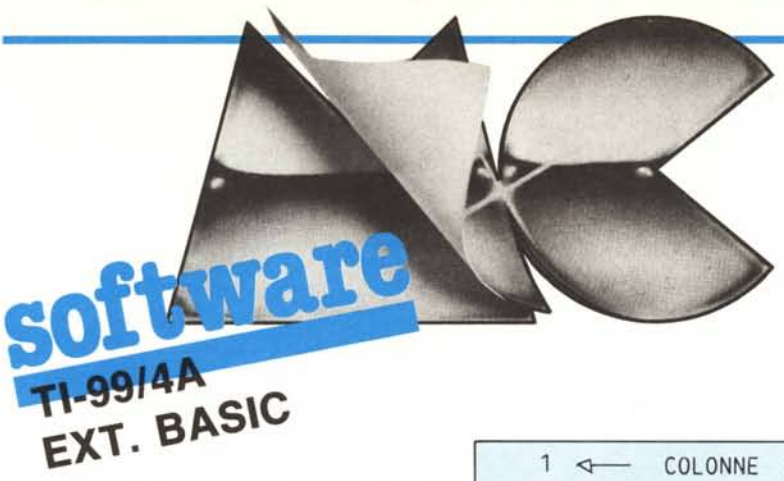


Funzioni per la grafica in alta risoluzione

di Carlo Randone - Chivasso (TO)

Ritorniamo a parlare di grafica, autentica croce e delizia degli utenti del TI-99/4A, per proporvi non la solita routine di indirizzamento del singolo pixel o il consueto programma per il "plottaggio" dei grafici di funzioni di una variabile, ma per presentarvi una libreria completa di routine grafiche in grado di aggiungere al vostro Basic 10 nuovi comandi. Risparmiandovi per questa volta la consueta tirata sulle limitazioni per quanto riguarda la grafica in alta risoluzione del Basic del TI 99, passiamo subito a introdurre l'argomento.

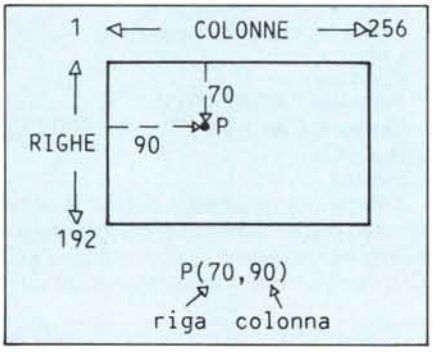
Il programma aggiunge al TI-99/4A le seguenti funzioni (richiamabili attraverso delle "CALL"):

N° d'ordine	Funzione	N° di linea
1	SAVEPIC1	(30060-30090)
2	SAVEPIC2	(30100-30240)
3	CIRCLE	(30250-30280)
4	TRAIL	(30290-30510)
5	PIXEL	(30520-30640)
6	TRIANG	(30650-30660)
7	RETTANG1	(30670-30730)
8	RETTANG2	(30740-30760)
9	MGLOBAL	(30770-30990)
10	CIRCLE3	(31000-31070)

Le subroutine, nel loro insieme, occupano circa 4,5 K di memoria e sono costituite da un totale di 108 linee di programma. Il loro scopo è quello di rendere disponibile all'utente una serie di istruzioni che permettono, in modo semplice e chiaro, di utilizzare la massima risoluzione grafica del nostro "Home".

Premesse

Ai fini dell'uso delle routine grafiche, lo schermo del monitor (o del televisore) viene così considerato:



Ogni pixel (minima unità video singolarmente indirizzabile) è individuato da una coppia di coordinate cartesiane ortogonali: la riga e la colonna (in QUESTO ordine!). Un generico punto del video deve avere coordinate di riga (R) e di colonna (C) tali che valgano le relazioni:

$1 \leqslant R \leqslant 192$

$1 \leqslant C \leqslant 256$

Siccome i caratteri ridefinibili, in Extended Basic, vanno dall'ASCII 33 all'ASCII 143 (escludiamo il 32 perché è necessario mantenere bianche le parti non disegnate), di conseguenza ne deriva che è possibile fare grafica in alta risoluzione utilizzando $(143-32) = 111$ matrici di 8×8 punti.

La disposizione di tali 111 caratteri sul video non è fissa, ma viene realizzata automaticamente, di volta in volta, dalle routine grafiche. Se si tenta di disegnare quando il calcolatore ha ormai ridefinito il 143° carattere, l'esecuzione della routine grafica in corso viene interrotta e viene emesso un segnale acustico, lo stesso che si avverte nel caso in cui si cerchi di indirizzare un pixel fuori dalla matrice 192×256 che costituisce il video. In totale, quindi, i punti singolarmente indirizzabili sono $(111 \times 64) = 7104$, purché non siano distribuiti su più di 111 matrici 8×8. In ogni caso, non ci sono problemi se ci si limita ad operare in alta risoluzione su un rettangolo video di 10×11 caratteri. Naturalmente, è possibile "disperdere" il disegno su tutto il video, ma, in questo caso, bisogna fare attenzione a non interessare più di 111 caratteri.

Non è possibile miscelare, se non parzialmente, testo e grafica (a causa della ridefinizione dei pattern dei caratteri). Restano utilizzabili le solite CALL COLOR e CALL SCREEN.

N.B. NON SI DEVE UTILIZZARE ASSOLUTAMENTE L'ISTRUZIONE "OPTION BASE 1", ma occorre lasciare il valore di default (che è OPTION BASE 0).

PONETE ATTENZIONE al fatto che LA PRIMA VOLTA (e solamente LA PRIMA) che viene chiamata una funzione grafica qualsiasi (escluse le N° 1, 2, 9), il puntatore dei DATA resta (a causa di un RESTORE presente nella funzione PIXEL) alla linea 30630 delle routine grafiche. Perciò, se il programma dell'utente prevede la lettura di istruzioni DATA (ossia, il caricamento di valori da delle DATA), è necessario che in quest'ultimo, *dopo* la prima chiamata ad una qualsiasi funzione grafica, sia presente un RESTORE seguito dall'eventuale numero di linea.

Vediamo ora quali sono le possibilità offerte dalle singole funzioni implementate.

Funzione: "PIXEL"

Formato: CALL PIXEL(R,C)

R = espressione, variabile o numero, tale che $1 \leqslant R \leqslant 192$

C = espressione, variabile o numero, tale che $1 \leqslant C \leqslant 192$

Finalità

Pone in stato "ON" il pixel situato nella posizione video di coordinate R,C (R = riga, C = colonna). Non necessariamente le coordinate devono essere degli interi, in quanto la funzione esegue automaticamente l'arrotondamento all'unità.

I punti indirizzabili sono completamente svincolati dalla "solita" scacchiera 24×32.

La routine esegue, ogni volta, due controlli:

— che le coordinate rispettino le relazioni $1 \leqslant R \leqslant 192$ e $1 \leqslant C \leqslant 256$;

— che non si tenti di ridefinire il pattern di un carattere con numero di codice superiore a 143, ossia che non si tenti di continuare ad indirizzare punti anche dopo aver utilizzata la 111ª matrice 8×8 disponibile.

Se queste condizioni non sono entrambe verificate, si ha l'emissione di un segnale acustico e l'uscita dalla funzione PIXEL.

La routine "PIXEL" è utilizzata da tutte le altre routine grafiche (tranne le SAVEPIC e la MGLOBAL), per cui nessuna delle altre funzioni grafiche è in grado di funzionare se non è presente in memoria la routine "PIXEL". Quest'ultima è però in grado di svolgere il suo compito indipendentemente dalla presenza o meno in memoria delle altre funzioni grafiche.

Esempio

Indirizzamento casuale di 100 punti sul video.

```
 1 CALL CLEAR :: RANDOMIZE :: CALL
SCREEN(2):: FOR H=1 TO 14 :: CALL COLOR
(H,H+2,1) :: NEXT H
 2 FOR I=1 TO 100
 3 CALL PIXEL(INT(RND*192)+1,INT
(RND*256)+1) :: CALL SOUND(10,1000,0)
 4 NEXT I
 5 GOTO 5
```

Funzione : "TRAIL"

Formato : CALL TRAIL(R1,C1,R2,C2)

Finalità

Disegna un segmento di retta fra il punto di coordinate R1,C1 ed il punto di coordinate R2,C2.

Esempio

Disegno di due segmenti che si incrociano.

```
 1 CALL CLEAR :: CALL SCREEN(2)
 2 FOR I=1 TO 14 :: CALL COLOR(I,I+2,1) ::
NEXT I
 3 CALL TRAIL(100,100,130,160)
 4 CALL TRAIL(70,130,150,100)
 5 GOTO 5
```

Funzione : "TRIANG"

Formato : CALL TRIANG(R1,C1,R2, C2,R3,C3)

Finalità

Disegna un triangolo date le coordinate dei suoi tre vertici: v1 (R1, C1); v2 (R2, C2); v3 (R3, C3).

Esempio

Disegno di due triangoli incrociati.

```
 1 CALL CLEAR :: CALL SCREEN(2) :: FOR
I=1 TO 14:: CALL COLOR (I,4,1) :: NEXT I
 2 CALL TRIANG(120,120,140,140,140,100)
 3 CALL TRIANG(120,100,120,140,160,120)
 4 GOTO 4
```

Funzione : "CIRCLE"

Formato : CALL CIRCLE(R,C,raggio)

Finalità

Disegna una circonferenza con il centro posto nel punto di coordinate R,C e con il raggio pari al valore specificato. Valori del raggio superiori a 30/40 utilizzano molti caratteri, attenzione perciò alle limitazioni conseguenti. Il raggio, comunque, deve avere valori misurati nella stessa scala delle coordinate.

Esempio

Disegno di 2 circonferenze tangenti.

```
1 CALL CLEAR
2 CALL CIRCLE(100,100,35)
3 CALL CIRCLE(100,155,20)
4 GOTO 4
```

Funzione: "CIRCLE3"

Formato: CALL CIRCLE3(R1,C1, R2,C2,R3,C3)

Finalità

Disegna una circonferenza date le coordinate di 3 suoi punti (per i quali, cioè, deve passare):

P1(R1,C1)
P2(R2,C2)
P3(R3,C3)

Attenzione a non dare le coordinate di 3 punti allineati!

Esempio

Disegno di un triangolo e del cerchio circoscritto.

```
 1 CALL CLEAR
 2 RA=120 :: CA=120
 3 RB=140 :: CB=140
 4 RC=140 :: CC=100
 5 CALL TRIANG(RA,CA,RB,CB,RC,CC)
 6 CALL CIRCLE3(RA,CA,RB,CB,RC,CC)
 7 GOTO 7
```

Funzione: "RETTANG1"

Formato: CALL RETTANG1(RV, CV,B,H,angolo)

Finalità

Disegna un rettangolo, di base B ed altezza H (misurate nella stessa scala delle coordinate), con il vertice V posto nel punto di coordinate RV e CV, e con la base ruotata rispetto all'orizzontale dell'angolo α ($0° \leqslant \alpha \leqslant 360°$), espresso in gradi sessagesimali.

Esempio

Disegno delle successive posizioni assunte da un rettangolo che ruota attorno ad un suo vertice, con incrementi angolari pari a 30° sessagesimali.

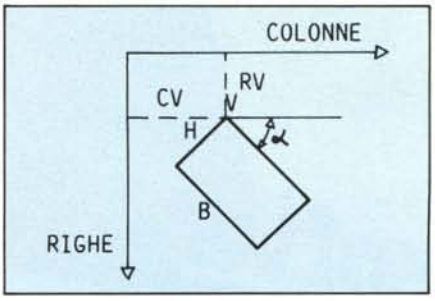
```
 1 CALL CLEAR :: CALL SCREEN(2)
 2 FOR I=1 TO 14 :: CALL COLOR(I,4,1) ::
NEXT I :: B=30 :: H=18
 3 FOR A=0 TO 330 STEP 30
 4 CALL RETTANG1(96,128,B,H,A)
 5 NEXT A
 6 GOTO 6
```

Funzione: "RETTANG2"

Formato: CALL RETTANG2(RO,CO, B,H,angolo)

Finalità

Disegna un rettangolo, di base B ed altezza H, con il centro (incontro delle diagonali) posto nel punto O, di coordinate RO, CO, e con la base ruotata rispetto all'orizzontale dell'angolo α ($0° \leqslant \alpha \leqslant 360°$), espresso in gradi sessagesimali.



Esempio

Disegno delle successive posizioni assunte da un rettangolo che ruota attorno a se stesso, con incrementi angolari pari a 30° sessagesimali.

```
 1 CALL CLEAR :: CALL SCREEN(2)
 2 FOR I=1 TO 14 :: CALL COLOR(I,4,1) ::
NEXT I :: B=30 :: H=18
 3 FOR A=0 TO 150 STEP 30
 4 CALL RETTANG2(96,128,B,H,A)
 5 NEXT A
 6 GOTO 6
```

Funzione: "MGLOBAL"

```
30000 !**********************
30010 !* Routine  grafiche *
30020 !* by__Randone Carlo *
30030 !**********************
30040 !
30050 !
30060 SUB SAVEPIC1
30070 PRINT :"Inserire la cassetta su cui" :: PRINT " registrare il disegno." ::
 PRINT "Prendere nota della posi-" :: PRINT " zione del nastro !!":
30080 OPEN #1:"CS1",INTERNAL,OUTPUT,FIXED(192)
30090 PRINT :: SUBEND
30100 SUB SAVEPIC2
30110 I=0 :: N=0 :: DIM NP(111),AA$(111)
30120 FOR R=1 TO 24 :: FOR C=1 TO 32
30130 CALL GCHAR(R,C,CAR):: IF CAR=32 THEN 30160
30140 N=N+1 :: CALL CHARPAT(CAR,AA$(N))
30150 NP(N)=(R-1)*32+C
30160 NEXT C :: NEXT R :: PRINT #1:N
30170 FOR T=1 TO N
30180 I=I+1 :: IF I=7 THEN I=0 :: GOTO 30210
30190 PRINT #1:NP(T),AA$(T),
30200 GOTO 30220
30210 PRINT #1:NP(T),AA$(T)
30220 NEXT T
30230 CALL SOUND(400,500,1):: CALL SOUND(600,900,1)
30240 SUBEND
30250 SUB CIRCLE(R_1,C_0,R)
30260 K=.5 :: RI=INT(R_1+K):: CO=INT(C_0+K)
30270 DI=1/R :: D=2*PI :: FOR A=0 TO D STEP DI :: CALL PIXEL(RI+R*COS(A),CO+R*SI
N(A)):: NEXT A
30280 SUBEND
30290 SUB TRAIL(Y_1,X_1,Y_2,X_2)
30300 A=1 :: K=.5 :: Y1=INT(Y_1+K):: X1=INT(X_1+K):: Y2=INT(Y_2+K):: X2=INT(X_2+
K)
30310 IF X1=X2 AND Y1=Y2 THEN 30440
30320 IF Y1=Y2 THEN 30460
30330 IF X1=X2 THEN 30490
30340 M=(Y2-Y1)/(X2-X1):: N=Y1-M*X1
30350 PX=ABS(X2-X1)
30360 PY=ABS(Y2-Y1)
30370 IF PX>=PY THEN 30410
30380 IF Y2<Y1 THEN A=-1
30390 FOR Y=Y1 TO Y2 STEP A :: X=(Y-N)/M :: CALL PIXEL(Y,X):: NEXT Y
30400 SUBEXIT
30410 IF X2<X1 THEN A=-1
30420 FOR X=X1 TO X2 STEP A :: Y=M*X+N :: CALL PIXEL(Y,X):: NEXT X
30430 SUBEXIT
30440 CALL PIXEL(Y1,X1)
30450 SUBEXIT
30460 IF X2<X1 THEN A=-1
30470 FOR X=X1 TO X2 STEP A :: CALL PIXEL(Y1,X):: NEXT X
30480 SUBEXIT
30490 IF Y2<Y1 THEN A=-1
30500 FOR Y=Y1 TO Y2 STEP A :: CALL PIXEL(Y,X1):: NEXT Y
30510 SUBEND
30520 SUB PIXEL(R_I,C_O)
30530 IF V<>0 THEN 30550
30540 DIM Y(22),W$(15,4):: GOTO 30590
30550 R=INT(R_I+KC):: C=INT(C_O+KC):: IF R<1 OR R>192 OR C<1 OR C>256 THEN 30640
30560 A=INT(R/G+H):: B=INT(C/G+H):: Z=C/G+H :: E=N*R-L*A+P+INT(Z/B):: CALL GCHAR
(A,B,K):: CALL CHARPAT(K,A$):: IF K=I THEN K,D=D+J
30570 IF D>143 THEN 30640
30580 CALL CHAR(K,SEG$(A$,J,E-J)&W$(Y(ASC(SEG$(A$,E,J))-M),C-G*INT(Z)+G)&SEG$(A$
,E+J,L)):: CALL HCHAR(A,B,K):: SUBEXIT
```

Formato: CALL MGLOBAL(C1,C2,C3,C4,C5)

Finalità

Effettua lo spostamento "globale" di tutto ciò che è presente sul video, sia esso testo o disegno, in bassa od alta risoluzione.

C1: è il codice ASCII del tasto che si vuole abilitare per lo spostamento in BASSO del quadro video.

C2: è il codice ASCII del tasto che si vuole abilitare per lo spostamento in ALTO del quadro video.

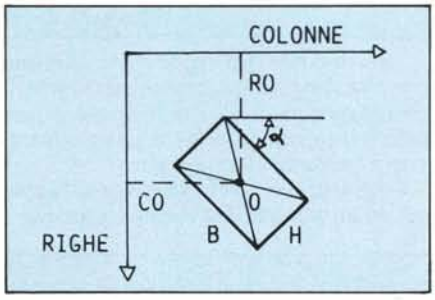
C3: è il codice ASCII del tasto che si vuole abilitare per lo spostamento a SINISTRA del quadro video.

C4: è il codice ASCII del tasto che si vuole abilitare per lo spostamento a DESTRA del quadro video.

C5: è il codice ASCII del tasto che si vuole abilitare per uscire dalla subroutine MGLOBAL e tornare al controllo del programma principale.

La MGLOBAL, quando chiamata, emette un segnale acustico di "READY", ed attende che si prema il tasto corrispondente alla funzione desiderata. Quindi effettua la traslazione (per lo spazio di 8 pixel = un carattere) nella direzione voluta, di tutto ciò che è rappresentato sul video.

I caratteri che eventualmente "uscissero" dal video, rientrano dalla corrispondente parte opposta. Al termine dello spostamento di tutti i 768 caratteri del video (tempo di esecuzione ≃ 30 sec.), si ha l'emissione di un segnale acustico e la routine resta nuovamente in attesa di comandi.



Esempio

Disegno di un rettangolo (normale e ruotato di 90 gradi), suo possibile spostamento sul video, disegno di un triangolo.

```
 1 CALL CLEAR :: CALL SCREEN(2)
 2 FOR I=1 TO 14 :: CALL COLOR(I,4,1) ::
NEXT I
 3 FOR A=0 TO 90 STEP 90
 4 CALL RETTANG2(96,128,80,50,A)
 5 NEXT A
 6 CALL MGLOBAL (69,88,83,68,ASC("P"))
 7 CALL TRIANG(96,128,128,96,132,132)
 8 GOTO 8
```

Il disegno dei due rettangoli può essere spostato usando i tasti con le frecce; per passare al disegno del triangolo, premere "P".

Funzione: "SAVEPIC1"

Formato: CALL SAVEPIC1 (è necessario far seguire una CALL CLEAR)

Finalità

Serve ad inizializzare il calcolatore ad una successiva memorizzazione su nastro del disegno in alta risoluzione che verrà eseguito. L'istruzione CALL SAVEPIC1 deve essere data da programma ed, in ogni caso, PRIMA DI INIZIARE IL TRACCIAMENTO DEL DISEGNO SUL VIDEO. Il calcolatore, trovando tale comando, richiede l'introduzione di un nastro libero nel 1° registratore a cassette (CS1), apre un file, e si dispone a memorizzare i dati del disegno, che gli saranno poi forniti dalla funzione SAVEPIC2 (vedi più avanti), che deve necessariamente essere presente (al termine della realizzazione del disegno), se è presente la SAVEPIC1.

Esempio

Vedi "SAVEPIC2".

Funzione: "SAVEPIC2"

Formato: CALL SAVEPIC2

Finalità

Tale funzione memorizza su nastro il disegno presente sul video. Non chiude il file, per evitare scorrimenti od alterazioni del disegno (a causa del messaggio di "STOP" del registratore conseguente alla chiusura di un file). Il file sarà chiuso automaticamente quando il calcolatore, nel programma realizzato dall'utente, troverà l'istruzione "END".

L'istruzione CALL SAVEPIC2 va scritta DOPO l'ultima istruzione di tracciamento del disegno; necessita dell'inizializzazione fornitagli dalla SAVEPIC1.

A memorizzazione terminata dei dati del disegno (posizione caratteri e loro identificazione di sagoma), la SAVEPIC2 emette un segnale acustico e ripassa il controllo al programma realizzato dall'utente.

Esempio

Tracciamento di due rettangoli e memorizzazione del disegno su nastro.

```
 1 CALL CLEAR
 2 CALL SAVEPIC1 :: CALL CLEAR
 3 CALL RETTANG2(100,100,40,25,10)
 4 CALL RETTANG2(50,130,20,30,30)
 5 CALL SAVEPIC2
 6 END
```

ATTENZIONE A NON REGISTRA-

```
30590 FOR I=0 TO 9 :: Y(I)=I :: NEXT I :: FOR I=17 TO 22 :: Y(I)=I-7 :: NEXT I :
: RESTORE 30620 :: D=32 :: V=1 :: Q=4 :: P=14 :: KC=.5
30600 FOR I=1 TO 4 :: FOR J=0 TO 15 :: READ B$ :: W$(J,I)=B$ :: NEXT J :: NEXT I
 :: G=8 :: H=.9 :: I=32 :: J=1 :: L=16 :: M=48 :: N=2
30610 GOTO 30550
30620 DATA 8,9,A,B,C,D,E,F,8,9,A,B,C,D,E,F,4,5,6,7,4,5,6,7,C,D,E,F,C,D,E,F
30630 DATA 2,3,2,3,6,7,6,7,A,B,A,B,E,F,E,F,1,1,3,3,5,5,7,7,9,9,B,B,D,D,F,F
30640 CALL SOUND(-50,860,0):: SUBEND
30650 SUB TRIANG(R1,C1,R2,C2,R3,C3)
30660 CALL TRAIL(R1,C1,R2,C2):: CALL TRAIL(R2,C2,R3,C3):: CALL TRAIL(R3,C3,R1,C1
):: SUBEND
30670 SUB RETTANG1(R_V,C_V,B,H,A1)
30680 K=.5 :: RV=INT(R_V+K):: CV=INT(C_V+K)
30690 IF A1<=90 THEN TV1=90-A1 ELSE TV1=450-A1
30700 TV3=TV1+270 :: TV1=TV1*PI/180 :: TV3=TV3*PI/180
30710 C1=CV+B*SIN(TV1):: R1=RV+B*COS(TV1):: C3=CV+H*SIN(TV3):: R3=RV+H*COS(TV3)
30720 C2=C3+B*SIN(TV1):: R2=R3+B*COS(TV1)
30730 CALL TRAIL(RV,CV,R1,C1):: CALL TRAIL(R1,C1,R2,C2):: CALL TRAIL(R2,C2,R3,C3
):: CALL TRAIL(R3,C3,RV,CV):: SUBEND
30740 SUB RETTANG2(R_O,C_O,B,H,A3)
30750 A2=A3*PI/180 :: K=.5 :: RO=INT(R_O+K):: CO=INT(C_O+K):: L=SQR(B*B+H*H)/2 :
: K2=A2+ATN(H/B)
30760 T0O=PI/2*3-K2 :: RO=RO+L*COS(T0O):: CO=CO+L*SIN(T0O):: CALL RETTANG1(RO,CO
,B,H,A3):: SUBEND
30770 SUB MGLOBAL(C1,C2,C3,C4,C5)
30780 IF VARIA()0 THEN 30800
30790 DIM A(32):: VARIA=1
30800 CALL SOUND(200,900,0)
30810 CALL KEY(0,K,S):: IF S=0 THEN 30810
30820 CALL SOUND(200,1000,0):: IF K=C1 THEN 30880
30830 IF K=C2 THEN 30910
30840 IF K=C3 THEN 30940
30850 IF K=C4 THEN 30970
30860 IF K=C5 THEN CALL SOUND(200,800,0):: SUBEXIT
30870 GOTO 30810
30880 FOR C=1 TO 32 :: CALL GCHAR(1,C,A(C)):: NEXT C
30890 FOR R=2 TO 24 :: FOR C=1 TO 32 :: CALL GCHAR(R,C,CAR):: CALL HCHAR(R-1,C,C
AR):: NEXT C :: NEXT R
30900 FOR C=1 TO 32 :: CALL HCHAR(24,C,A(C)):: NEXT C :: GOTO 30800
30910 FOR C=1 TO 32 :: CALL GCHAR(24,C,A(C)):: NEXT C
30920 FOR R=23 TO 1 STEP -1 :: FOR C=1 TO 32 :: CALL GCHAR(R,C,CAR):: CALL HCHAR
(R+1,C,CAR):: NEXT C :: NEXT R
30930 FOR C=1 TO 32 :: CALL HCHAR(1,C,A(C)):: NEXT C :: GOTO 30800
30940 FOR R=1 TO 24 :: CALL GCHAR(R,1,A(R)):: NEXT R
30950 FOR C=2 TO 32 :: FOR R=1 TO 24 :: CALL GCHAR(R,C,CAR):: CALL HCHAR(R,C-1,C
AR):: NEXT R :: NEXT C
30960 FOR R=1 TO 24 :: CALL HCHAR(R,32,A(R)):: NEXT R :: GOTO 30800
30970 FOR R=1 TO 24 :: CALL GCHAR(R,32,A(R)):: NEXT R
30980 FOR C=31 TO 1 STEP -1 :: FOR R=1 TO 24 :: CALL GCHAR(R,C,CAR):: CALL HCHAR
(R,C+1,CAR):: NEXT R :: NEXT C
30990 FOR R=1 TO 24 :: CALL HCHAR(R,1,A(R)):: NEXT R :: GOTO 30800 :: SUBEND
31000 SUB CIRCLE3(Y_1,X_1,Y_2,X_2,Y_3,X_3)
31010 K=.5 :: Y1=INT(Y_1+K):: X1=INT(X_1+K):: Y2=INT(Y_2+K):: X2=INT(X_2+K):: Y3
=INT(Y_3+K):: X3=INT(X_3+K):: GOTO 31060
31020 T1=-(X1*X1+Y1*Y1):: T2=-(X2*X2+Y2*Y2):: T3=-(X3*X3+Y3*Y3):: DE=X1*(Y2-Y3)-
Y1*(X2-X3)+X2*Y3-Y2*X3
31030 DA=T1*(Y2-Y3)-Y1*(T2-T3)+T2*Y3-T3*Y2 :: DB=X1*(T2-T3)-T1*(X2-X3)+X2*T3-X3*
T2
31040 DC=X1*(Y2*T3-Y3*T2)-Y1*(X2*T3-X3*T2)+T1*(X2*Y3-X3*Y2):: CALL PIXEL(Y1,X1):
: CALL PIXEL(Y2,X2):: CALL PIXEL(Y3,X3)
31050 A=DA/DE :: B=DB/DE :: C=DC/DE :: XC=-A/2 :: YC=-B/2 :: R=SQR(XC*XC+YC*YC-C
):: CALL CIRCLE(YC,XC,R):: SUBEXIT
31060 IF (X1=X2 AND X2=X3)OR(Y1=Y2 AND Y2=Y3)THEN SUBEXIT
31070 GOTO 31020 :: SUBEND
```

RE UN DISEGNO SOPRA IL PROGRAMMA DELLE FUNZIONI GRAFICHE!!!

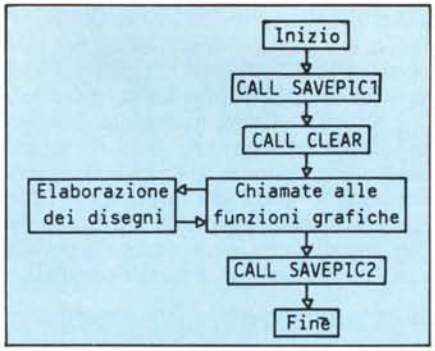
I disegni memorizzati su nastro (in forma codificata), possono essere LETTI (e riprodotti velocemente su video) tramite il programma di LETTURA DISEGNI.

Per disegni di una certa complessità è più rapida la visualizzazione tramite lettura da nastro che non tramite riscrittura e riesecuzione delle chiamate alle funzioni grafiche necessarie per la loro creazione!

È consigliabile memorizzare i disegni su di una cassetta diversa da quella contenente le funzioni grafiche; sul nastro di memorizzazione, per praticità, è opportuno registrare (a partire dalla posizione 000) il programma di lettura disegni. I file contenenti i vari disegni saranno registrati *dopo* tale programma (che serve a curarne e gestirne l'interpretazione).

Notate il fatto che le istruzioni SAVEPIC servono esclusivamente nel caso in cui si preveda di realizzare un disegno che si desidera poi conservare su nastro.

In ogni caso, lo schema per l'utilizzazione delle funzioni SAVEPIC, è il seguente:



Se si desidera (per risparmiare memoria o per limitare i tempi di lettura da nastro di un programma finito) sopprimere alcune delle funzioni/subroutine, si tenga attentamente presente quanto segue:

— se le routine grafiche vengono "appese" ad un programma che non necessita e non prevede di poter salvare su nastro i disegni, le funzioni SAVEPIC1, e SAVEPIC2 possono essere eliminate.

— Così può anche essere cancellata la MGLOBAL se non si ha la necessità di spostare disegni sul video.

— Per quanto riguarda l'eventuale eliminazione delle altre routine (più specificamente grafiche), si faccia riferimento alla tabella sottostante.

Il rettangolino nero significa: se si elimina la funzione in alto (riga "FUNZIONI utilizzate"), si deve eliminare anche la funzione a sinistra (colonna "FUNZIONE").

Esempi di applicazione della tabella:

1) Si può eliminare la subroutine RETTANG1 e lasciare attive tutte le altre?

Nella riga "FUNZIONI utilizzate" si cerca la RETTANG1. Spostandosi verso il basso, il primo ed unico rettangolo nero che si incontra è posto in corrispondenza della RETTANG2: questo vuol dire che la funzione RETTANG2 necessita della RETTANG1, per cui diventa anch'essa inutilizzabile eliminando la RETTANG1.

	FUNZIONI utilizzate						
FUNZIONE	CIRCLE	TRAIL	PIXEL	TRIANG	RETTANG1	RETTANG2	CIRCLE3
CIRCLE	■		■				
TRAIL		■	■				
PIXEL			■				
TRIANG		■	■	■			
RETTANG1		■	■		■		
RETTANG2		■	■		■	■	
CIRCLE3	■		■				■

2) Si può eliminare la subroutine pixel e lasciare attive tutte le altre?

Fra le "FUNZIONI utilizzate" cerchiamo la PIXEL. Spostandoci verso il basso vediamo che tutte le altre funzioni grafiche utilizzano la PIXEL, per cui non è possibile eliminarla.

3) Si può eliminare la subroutine TRAIL e lasciare attive le altre? Dalla tabella è possibile vedere come l'eliminazione della TRAIL comporta la caduta anche delle funzioni TRIANG, RETTANG1 e RETTANG2.

La velocità di esecuzione delle routine è tutto sommato soddisfacente. Un'unica osservazione da fare è che la funzione circle non disegna un cerchio perfetto ma una figura leggermente ellittica: questo fatto non è dovuto ad un cattivo funzionamento del programma, e neppure a difetti di regolazione del monitor, ma esclusivamente al fatto che i pixel del TI-99 non sono quadrati o circolari, ma rettangolari! Volendo, chi lo desidera può ridefinire la funzione in modo tale da compensare lo schiacciamento del cerchio, per esempio sostituendo l'equazione della circonferenza (quella del programma è in forma parametrica) con quella di una ellisse con l'asse minore parallelo alle righe del video (ossia orizzontale). Si tenga presente che il rapporto fra il lato minore di un pixel (rettangolare) ed il lato maggiore, vale circa 0.71. MC

Lettore di disegni

```
10 CALL CLEAR :: COL=2 :: SCH=2 :: DISPLAY AT(1,1):"------------------------------
"
20 DISPLAY AT(2,1):"LETTURA DISEGNI DA NASTRO"
30 DISPLAY AT(3,1):"------------------------------"
40 DISPLAY AT(5,1):"I disegni dovresti averli" :: DISPLAY AT(6,1):"memorizzati s
u questo stesso" :: DISPLAY AT(7,1):"nastro, dopo il presente"
50 DISPLAY AT(8,1):"programma principale."
60 DISPLAY AT(10,1):"A DISEGNO TERMINATO, PREMI :" :: DISPLAY AT(11,1):" C per c
ambiarne il colore;"
70 DISPLAY AT(12,1):" S per cambiare il colore" :: DISPLAY AT(13,1):"   dello sf
ondo;"
80 DISPLAY AT(14,1):" UN ALTRO TASTO per finire."
90 DISPLAY AT(16,1):"COD. COLORE SFONDO :" :: ACCEPT AT(16,22)BEEP:CS
100 DISPLAY AT(18,1):"COD. COLORE DISEGNO :"
110 DISPLAY AT(20,1):" COLORE UNICO (U) O COLORATO"
120 DISPLAY AT(22,1):" CON VARIE TINTE (V) ?" :: ACCEPT AT(22,24)VALIDATE("UV")B
EEP:Z$
130 IF Z$="V" THEN 150
140 DISPLAY AT(24,1):"COD. COLORE DISEGNO :" :: ACCEPT AT(24,23)BEEP:CD
150 PRINT :: OPEN #1:"CS1",INTERNAL,INPUT ,FIXED(192)
160 CALL CLEAR :: CALL SCREEN(CS):: IF Z$="V" THEN 170 ELSE 180
170 FOR I=1 TO 14 :: CALL COLOR(I,I+2,1):: NEXT I :: GOTO 190
180 FOR I=1 TO 14 :: CALL COLOR(I,CD,1):: NEXT I
190 I=0 :: H=32 :: INPUT #1:N
200 FOR Y=1 TO N :: I=I+1 :: IF I=7 THEN I=0 :: GOTO 230
210 INPUT #1:NP1,A$,
220 GOTO 240
230 INPUT #1:NP1,A$
240 H=H+1 :: CALL CHAR(H,A$):: R=INT(NP1/32+.97):: CALL HCHAR(R,NP1-(R-1)*32,H):
: NEXT Y
250 CALL SOUND(400,500,1):: CALL SOUND(600,900,1)
260 CALL KEY(0,K,S):: IF S=0 THEN 260
270 IF CHR$(K)="C" THEN 380
280 IF CHR$(K)="S" THEN VRR=1 :: GOTO 450
290 CALL CLEAR :: CALL CHARSET :: FOR I=1 TO 14 :: CALL COLOR(I,2,1):: NEXT I ::
 CALL SCREEN(8):: CLOSE #1
300 CALL CLEAR :: PRINT "1-USCIRE DAL PROGRAMMA"
310 PRINT :"2-LEGGERE UN ALTRO DISEGNO"
320 PRINT : : : : : : :
330 CALL KEY(0,K,S):: IF S=0 THEN 330
340 IF CHR$(K)="1" THEN 370
350 IF CHR$(K)="2" THEN 10
360 GOTO 330
370 CALL CLEAR :: END
380 COL=COL+1 :: I=0 :: IF COL>16 THEN COL=2
390 I=I+1 :: CALL COLOR(I,COL,1)
400 CALL KEY(0,K,S):: IF S=0 THEN 400
410 IF CHR$(K)="C" THEN 440
420 IF CHR$(K)="S" THEN VRR=2 :: GOTO 450
430 GOTO 290
440 FOR TYY=1 TO 20 :: NEXT TYY :: IF I=14 THEN 260 ELSE 390
450 SCH=SCH+1 :: IF SCH>16 THEN SCH=2
460 CALL SCREEN(SCH):: FOR TYY=1 TO 20 :: NEXT TYY
470 ON VRR GOTO 260,400
```

Elenco del software disponibile su cassetta o minifloppy

Per ovviare alle difficoltà incontrate da molti lettori nella digitazione dei listati pubblicati nelle varie rubriche di software sulla rivista, MCmicrocomputer mette a disposizione i programmi più significativi direttamente su supporto magnetico. Riepiloghiamo qui a fianco i programmi disponibili per le varie macchine, ricordando che i titoli non sono previsti per computer diversi da quelli indicati.
Il numero della rivista su cui viene descritto ciascun programma è riportato nell'apposita colonna; consigliamo gli interessati di procurarsi i relativi numeri arretrati, eventualmente rivolgendosi al nostro Servizio Arretrati utilizzando il tagliando pubblicato in fondo alla rivista.

Per l'ordinazione inviare l'importo (a mezzo assegno, c/c o vaglia postale) alla Technimedia srl, Via Valsolda 135, 00141 Roma.

Le cassette utilizzate sono Basf C-60 Compusette II; i minifloppy sono Basf singola faccia singola densità.

Codice	Titolo programma	MC n.	Prezzo	Note
	APPLE II			
DA2/00	Shape Tablet	22	15000	
DA2/01	Motomuro	26	15000	
DA2/02	&DEBUG	28	15000	
DA2/03	EDIT + INPUT	29	15000	
DA2/04	Basic modulare	34	15000	
DA2/05	ANNA Animation Lang.	35/37	15000	
DA2/06	Miniset + Leva-DOS	37	15000	
DA2/07	27 programmi grafici	38	30000	
DA2/08	Adventure Editor	38	15000	
	COMMODORE 64			
C64/01	Briscola	25	17000	
C64/02	Serpentone	29	17000	
C64/03	Othello	29	17000	
C64/04	Chase	33	17000	
C64/05	Spreadsheet	34	30000	
C64/06	Bilancio familiare	35	17000	
C64/07	The dark wood	36	17000	
C64/08	Totocalcio: sis.rid.	37	17000	
C64/09	Orchetes	37	17000	
C64/10	Wordprocessor	38	17000	
C64/11	Helicopt	38	17000	
C64/12	Finestra grafica	39	17000	
C64/13	Paroliamo	39	17000	
C64/14	Scarabeo	40	17000	
D64/01	Spreadsheet	34	15000	
D64/02	ADP Basic	da 35 a 39	15000	
D64/03	Wordprocessor	38	15000	
D64/04	Paroliamo	39	15000	
D64/05	Data base Galileo	40/41	15000	
	COMMODORE VIC-20			
CVC/01	VIC-Maze	19	17000	Config. base
CVC/02	Pic-Man	23	17000	Config. base
CVC/03	Briscola	25	17000	Config. base
CVC/04	Grand Prix	28	17000	Config. base
CVC/05	Frogger	26	17000	RAM: almeno + 3 K
CVC/06	Invaders	29	23000	RAM: + 16 K
CVC/07	Othello	29	17000	RAM: + 16 K
CVC/08	SKI	31	17000	Config. base
CVC/09	VIC-quiz	32	17000	RAM: almeno + 8 K
CVC/10	Zigurat	33	17000	Config. base
CVC/11	Extended Basic	36	17000	RAM: + 16 K
CVC/12	Fireman	36	17000	Config. base
CVC/13	Accordi per chitarra	39	17000	RAM: almeno + K
CVC/14	Piramide di Iunnuh	39	17000	RAM: almeno + K
CVC/15	Il castello	40	17000	RAM: + 16 K
DVC/01	EXMA	27/28	15000	RAM: + 16 K
	SINCLAIR SPECTRUM			
CSS/01	TRILAB	28	17000	
CSS/02	SET di caratteri	27/29	17000	
CSS/03	Grafica TREDIM	29	17000	
CSS/04	Ippica	30	17000	
CSS/05	Graphic-Comp	32	17000	
CSS/06	Macchina del tempo	34	17000	
CSS/07	Piramide di Iunnuh	35	17000	
CSS/08	Over Basic	37	17000	
CSS/09	Prospettiva	38	17000	
CSS/10	Motomuro	39	17000	48 K RAM
CSS/11	Othello	40	17000	
CSS/12	The dark wood	40	17000	
	TEXAS TI-99/4A			
CT9/01	Macchina del tempo	27	17000	
CT9/02	Simon	29	17000	
CT9/03	Babilonia	30	17000	
CT9/04	Labirinto 3D	31	17000	
CT9/05	Piramide di Iunnuh	33	17000	Extended Basic
CT9/06	Scrabble	34	17000	
CT9/07	Morphy	35	17000	
CT9/08	Equo canone	37	17000	
CT9/09	Scopa	39	17000	
CT9/10	Montecarlo	39	17000	Extended Basic

Nota:
l'iniziale del codice e' C per le cassette, D per i minifloppy