

Questo mese presentiamo un simpatico programma per disegnare istogrammi sulla base dei dati che introduciamo da tastiera, che naturalmente permette anche il salvataggio e/o la stampa dei grafici ottenuti. Promosso a pieni voti per la facilità operativa, questo programma ha una grafica molto curata, è sufficientemente veloce ed è scritto in un AmigaBasic molto pulito: menu a discesa, gestione degli interrupt, niente GOTO ecc.

Nonostante la sua «proibitiva» lunghezza pubblichiamo ugualmente il listato per intero, proprio per il suo valore che non esitiamo a definire «didattico». Complimenti.

Ma la caratteristica certamente più interessante di tutto il programma, è l'uso delle librerie di sistema da Basic per poter utilizzare font diversi da quello standard e ottenere la stampa dei grafici su carta. Con la promessa di tornare presto sull'argomento (librerie, non grafici!) cominciate col gustarvi questa preziosa (e comoda) utility.

Buon divertimento

a d p

È disponibile, presso la redazione, il disco con il programma pubblicato in questa rubrica. Le istruzioni per l'acquisto e l'elenco degli altri programmi disponibili sono a pag. 249.

## Grafici

di Marco Sivori, Bolzaneto (GE)

Questo programma nasce dall'esigenza di avere una utility di facilissimo impiego e molto veloce per realizzare grafici per poi poterli stampare su carta.

Da questo software si possono ottenere istogrammi verticali, orizzontali e «più e meno», per qualsiasi esigenza, ed è completamente gestito col mouse di modo che gli unici accessi alla tastiera siano limitati all'inserimento dei dati (torna molto comodo, a questo scopo, utilizzare il pad numerico). Chi acquisterà il programma su dischetto, troverà anche 3 file di dati dimostrativi (GASOLIO, COMPUTER e MAGAZZINI) che ne illustrano il funzionamento, nonché la versione compilata (grafi-run) del programma il quale, oltre ad incrementarne la velocità, evita di «portarsi dietro» oltre 100 K di interprete.

Il programma si presenta con una schermata grafica che mostra i tre tipi di grafico rappresentabile, quindi...

### Funzionamento del programma

Utilizzando il mouse, si hanno due possibilità di scelta: inserire i dati o caricare un file salvato precedentemente. Nel primo caso, nel menu PROGETTO, optare per la richiesta NUOVO. Si aprirà una *window*. Il primo dato da inserire è relativo al numero degli inserimenti (ne sono consentiti come minimo uno e come massimo tredici; sono sufficienti per fare un diagramma del tipo «mensilizzazione budget e media mensile»), quindi vanno inseriti i dati veri e propri. Dopodiché apparirà la richiesta del nome del grafico. Questo input può accettare sia stringhe che numeri (ad esempio «Budget 1988»). Quindi inseri-

re le label (o etichette) che dovranno fare riferimento a ciascun dato precedentemente inserito.

Terminata questa fase, spostarsi con il mouse sul menu TIPO e scegliere il tipo di grafico da rappresentare. Immediatamente verrà visualizzato sullo schermo. Dal menu PROGETTO scegliere STAMPA e si otterrà l'hard copy su carta. Quindi SALVA e/o FINE per terminare. Tutto qua...

### Descrizione particolareggiata

#### MENU PROGETTO

**NUOVO.** Serve per inserire i dati da tastiera. L'ordine di inserimento è: dati numerici, titolo del grafico ed etichette (stringhe). Nel caso dati siano stati già inseriti precedentemente, premendo ENTER (sulla tastiera o sul pad numerico) si confermerà il dato precedente (ad eccezione del titolo del grafico che bisognerà reinserire). Ad esempio, se erano stati inseriti dieci valori e si volesse rappresentare un grafico solo con tre, digitare 3, premere ENTER tre volte, inserire il titolo, premere ENTER ancora tre volte. Nel caso i dati da visualizzare fossero completamente diversi, reinserirli.

**CARICA.** Carica i dati da disco. Alla richiesta del nome del file da caricare, se si preme ENTER senza aver inserito alcun carattere, si uscirà da questa fase. In pratica consente di annullare questo comando. **Importante:** il nome del file da caricare deve essere digitato senza l'estensione .barre (che provvederà il computer a mettere).

**SALVA.** Salva i dati su disco. In questa fase non è necessario specificare nessun nome per il file di dati. Verranno salvati automaticamente con il nome grafico.barre.

**SALVA COME.** Salva i dati su disco con

```

'GRAFICI
'M & F SIVORI - 1.88
DEFIMG a=2
DIM dato(12), labels(12), ok(12)
LIBRARY "diskfont.library"
LIBRARY "exec.library"
LIBRARY "graphics.library"
DECLARE FUNCTION OpenDiskFont LIBRARY
DECLARE FUNCTION AllocSignal%() LIBRARY
DECLARE FUNCTION AllocMem%() LIBRARY
DECLARE FUNCTION FindTask%() LIBRARY
DECLARE FUNCTION DoIO%() LIBRARY
DECLARE FUNCTION OpenDevice& LIBRARY
dati=0 'nessun dato ancora inserito

inizio:
SCREEN 2,320,250,4,1
WINDOW 2,"GRAFICI",,16,2
tipo=1 'grafico di default (quello verticale)
caratteres="ruby.font"
PALETTE 0,1,1,1
PALETTE 1,0,0,0
GOSUB colori

MENU 1,0,1,"PROGETTO"
MENU 1,1,1,"NUOVO"
MENU 1,2,1,"CARICA"
MENU 1,3,1,"SALVA"
MENU 1,4,1,"SALVA COME"
MENU 1,5,1,"STAMPA"
MENU 1,6,1,"FINE"
MENU 2,0,1,"GRIGLIA"
MENU 2,1,1,"ORIZZONTALE"
MENU 2,2,1,"VERTICALE"
MENU 2,3,1,"ELIMINA"
MENU 3,0,1,"PALETTE"
MENU 3,1,1,"COLORI"
MENU 3,2,1,"BIANCO E NERO"
MENU 4,0,1,"TIPO"
MENU 4,1,1,"VERT."
MENU 4,2,1,"ORIZZ."
MENU 4,3,1,"+ & -"

LINE(10,20)-(110,70),2,bf 'sfondi
LINE(10,90)-(110,140),2,bf
LINE(10,160)-(110,210),2,bf

LINE(20,60)-(90,60),1
LINE(20,100)-(20,130),1
LINE(20,185)-(90,185),1

LINE(30,60)-(40,30),3,bf 'verticali
LINE(50,60)-(60,50),4,bf
LINE(70,60)-(80,40),5,bf

LINE(20,100)-(100,105),6,bf 'orizzontali
LINE(20,115)-(40,120),7,bf
LINE(20,130)-(70,135),8,bf

LINE(30,185)-(40,170),9,bf 'verticali
LINE(50,185)-(60,200),10,bf
LINE(70,185)-(80,165),11,bf

size=15
GOSUB carica.font
LOCATE 1,4:COLOR 1:PRINT "GRAFICI A BARRE"
size=12:GOSUB carica.font
LOCATE 4,9:PRINT "VERTICALI"
LOCATE 10,9:PRINT "ORIZZONTALI"
LOCATE 16,9:PRINT "PIU' & MENO"
size=8:GOSUB carica.font

loop:
MENU ON
ON MENU GOSUB cerca.menu
GOTO loop

cerca.menu:
menu1=MENU(0)
menu2=MENU(1)
ON menu1 GOSUB progetto.griglia.paletta.tipo
RETURN

END

progetto:
ON menu2 GOSUB nuovo.carica.salva.salva.come.stampa.carta.fine
RETURN

griglia:
IF dati=0 THEN RETURN
ON menu2 GOSUB gr.or.gr.ve.gr.an
RETURN

paletta:
ON menu2 GOSUB colori.monocrom
RETURN

tipo:
ON menu2 GOSUB verticale.orizzontale.piu.meno
RETURN

'sotto sotto menu'-----

colori:
PALETTE 2,4,1,1
PALETTE 3,1,55,35
PALETTE 4,55,7,05
PALETTE 5,55,55,5
PALETTE 6,85,85,0
PALETTE 7,85,3,75
PALETTE 8,35,7,95
PALETTE 9,8,4,15
PALETTE 10,4,45,15
PALETTE 11,4,45,4
PALETTE 12,1,2,05
PALETTE 13,05,75,15
PALETTE 14,0,55,95
PALETTE 1,0,0,0
RETURN

monocrom:
PALETTE 2,25,25,25
PALETTE 3,65,65,65
PALETTE 4,4,4,4
PALETTE 5,5,5,5
PALETTE 6,63,65,45
PALETTE 7,48,5,45
PALETTE 8,48,55,5
PALETTE 9,6,6,6
PALETTE 10,3,3,25
PALETTE 11,55,45,0
PALETTE 12,5,5,5
PALETTE 13,45,45,45
PALETTE 14,3,4,4
PALETTE 1,0,0,0
RETURN

gr.or:
'griglia orizzontale
IF tipo=2 THEN
larg=(150/n-7)
FOR x=0 TO n-1
ok(x)=(ok(x)/max)*285 '285=massima lunghezza
NEXT x
interv=7
FOR x=0 TO n-1
LINE(5,interv+40+(larg/2))-(295,interv+40+(larg/2)),1
interv=interv+larg+7
NEXT x
END IF
IF tipo=1 OR tipo=3 THEN
FOR uu=50 TO 200 STEP 25
LINE(5,uu)-(295,uu),1
NEXT uu
END IF
RETURN

gr.ve:
'griglia verticale
IF tipo=1 OR tipo=3 THEN
larg=(290/n-10)
'calcola ampiezza massima di ogni barra
FOR x=0 TO n-1
ok(x)=(ok(x)/max)*150 '150=massima altezza
NEXT x
interv=10
FOR x=0 TO n-1
LINE(interv+(larg/2),205)-(interv+(larg/2),40),1
interv=interv+larg+10
NEXT x
END IF
IF tipo=2 THEN
FOR uu=5 TO 295 STEP 29
LINE(uu,205)-(uu,40),1
NEXT uu
END IF
RETURN

gr.an:
IF tipo=1 THEN GOSUB verticale
IF tipo=2 THEN GOSUB orizzontale
IF tipo=3 THEN GOSUB piu.meno
RETURN

nuovo:
MENU OFF
WINDOW 3,"INSERIMENTO DATI",(10,20)-(300,100),0,2
WINDOW OUTPUT 3
CLS:INPUT "Numero dati da inserire (1-13)":n:CLS
IF n<1 THEN n=1
IF n>13 THEN n=13
FOR x=0 TO n-1:LOCATE 2,9:PRINT "INSERIMENTO N.":x+1
LOCATE 4,1:PRINT "Precedente ":dato(x)
LOCATE 6,1:INPUT "NUOVO":ds
IF ds="" THEN dato(x)=dato(x) ELSE dato(x)=VAL(ds)
CLS
NEXT x
LOCATE 5,1:PRINT "TITOLO -----"
LOCATE 5,7:INPUT " ":titolo$
IF LEN(titulo$)>20 THEN titulo$=LEFT$(titulo$,20)
dati=1 'OK, Sono stati inseriti
CLS:PRINT "INSERIRE LE LABELS, MAX 4 CARATTERI."
FOR x=0 TO n-1:LOCATE 2,9:PRINT "INSERIMENTO N.":x+1
LOCATE 4,1:PRINT "Precedente ":labels(x)
LOCATE 6,1:INPUT "NUOVA ":labs$
IF labs="" THEN labels(x)=labels(x) ELSE labels(x)=labs$
IF LEN(labs$)>4 THEN labels(x)=LEFT$(labs$,4)
CLS
NEXT x
WINDOW OUTPUT 2:WINDOW CLOSE 3
RETURN

```

(continua a pagina 226)

il nome scelto dall'utente. Nella finestra dedicata a questa opzione, viene mostrato il nome del grafico in uso in quel momento; inserire il nuovo nome *omettendo l'estensione .barre*. Nel caso si preme ENTER senza aver digitato niente, si potrà uscire da questa opzione consentendo di dare un nuovo comando.

**STAMPA.** Stampa il grafico su carta. Previamente, si sarà provveduto a settare opportunamente la stampante con il programma PREFERENCES indicando il tipo posseduto ed i vari parametri (driver e tipo di stampa grafica). I migliori risultati, ovviamente, si ottengono con stampanti a colori. *Importante:* prima di dare

questo comando la stampante deve essere correttamente collegata ed accesa.

**FINE.** Esce dal programma, ma rimane in Basic per consentire di listare il programma.

Eventualmente è sufficiente inserire SYSTEM nell'apposita routine (fine:) per uscire al sistema operativo.

(continua da pagina 225)

```

carica:
MENU OFF
WINDOW 3,"CARICAMENTO DATI", (10,20)-(300,100),0,2
WINDOW OUTPUT 3
CLS
INPUT "NOME DEL FILE": nomeGRAF1$
IF nomeGRAF1$="" THEN WINDOW OUTPUT 2: WINDOW CLOSE 3: RETURN
nomeGRAF$=nomeGRAF1$+".barre"
CLS: PRINT: PRINT "ATTENDERE... CARICAMENTO IN CORSO."
ON ERROR GOTO errori:
n=0
OPEN nomeGRAF$ FOR INPUT AS #1
WHILE NOT EOF(1)
INPUT #1, dato(n)
INPUT #1, label$(n)
INPUT #1, titolo$
n=n+1
WEND
CLOSE #1
dati=1
WINDOW OUTPUT 2: WINDOW CLOSE 3
RETURN

errori:
IF ERR=53 THEN CLOSE #1: BEEP: BEEP: WINDOW OUTPUT 2: WINDOW CLOSE 3: RESUME
inizio
RESUME inizio

salva:
MENU OFF
IF dati=1 THEN
WINDOW 3,"SALVATAGGIO DATI", (10,20)-(300,100),0,2
WINDOW OUTPUT 3
CLS: PRINT: PRINT: PRINT "ATTENDERE... SALVATAGGIO IN CORSO."
OPEN "grafico.barre" FOR OUTPUT AS #1
FOR x=0 TO n-1
PRINT #1, dato(x)
PRINT #1, label$(x)
PRINT #1, titolo$
NEXT x
CLOSE #1
WINDOW OUTPUT 2: WINDOW CLOSE 3
END IF
RETURN

salva.come:
MENU OFF
IF dati=0 THEN RETURN
WINDOW 3,"SALVATAGGIO DATI", (10,20)-(300,100),0,2
WINDOW OUTPUT 3
CLS: PRINT: PRINT "NOME PREC.": nomeGRAF$: PRINT
INPUT "NOME DA ASSEGNARE": nomeGRAF1$
IF nomeGRAF1$="" THEN WINDOW OUTPUT 2: WINDOW CLOSE 3: RETURN
nomeGRAF$=nomeGRAF1$+".barre"
CLS: PRINT: PRINT: PRINT "ATTENDERE... SALVATAGGIO IN CORSO."
OPEN nomeGRAF$ FOR OUTPUT AS #1
FOR x=0 TO n-1
PRINT #1, dato(x)
PRINT #1, label$(x)
PRINT #1, titolo$
NEXT x
CLOSE #1
WINDOW OUTPUT 2: WINDOW CLOSE 3
RETURN

stampa.carta:
MENU OFF
IF dati=0 THEN RETURN
CALL stampa
RETURN

fine:
MENU RESET
SCREEN CLOSE 2
END
-----
verticale:
MENU OFF
IF dati<>1 THEN RETURN
CLS
tipo=1
FOR x=0 TO n-1
IF dato(x)<=0 THEN ok(x)=0 ELSE ok(x)=dato(x)
NEXT x
GOSUB calcola.massimo
'calcola ampiezza massima di ogni barra
larg=290/n-10
'calcola ampiezza massima di ogni barra
FOR x=0 TO n-1
ok(x)=(ok(x)/max)*150 '150=massima altezza

```

```

NEXT x
'10=intervallo tra una barra e l'altra - 200=base del grafico
interv=10
FOR x=0 TO n-1
LINE(interv,200)-(interv+larg,200-ok(x)),x+2,bf
LINE(interv,200)-(interv+larg,200-ok(x)),1,b
interv=interv+larg+10
NEXT x
'stampa coordinate
LINE(5,200)-(295,200),1 'linea orizzontale
LINE(5,200)-(5,40),1 'linea verticale
GOSUB stampa.titolo
carattere$="topaz.font":size=8:GOSUB carica.font
FOR x=0 TO INT((n-1)/2)
LOCATE 27,(x*5)+1:COLOR 1,x+2:PRINT label$(x)
NEXT x
x1=1
FOR x=INT(((n-1)/2)+1) TO n-1
LOCATE 28,x1:COLOR 1,x+2:PRINT label$(x)
x1=x1+5
NEXT x:COLOR 1,0
LOCATE 5,1:PRINT max
RETURN

```

```

orizzontale:
MENU OFF
IF dati<>1 THEN RETURN
CLS
tipo=2
FOR x=0 TO n-1
IF dato(x)<=0 THEN ok(x)=0 ELSE ok(x)=dato(x)
NEXT x
GOSUB calcola.massimo
'calcola ampiezza massima di ogni barra
larg=150/n-7
'calcola ampiezza massima di ogni barra
FOR x=0 TO n-1
ok(x)=(ok(x)/max)*285 '285=massima lunghezza
NEXT x
'7=intervallo tra una barra e l'altra - 200=base del grafico
interv=7
FOR x=0 TO n-1
LINE(5,interv+40)-(5+ok(x),interv+larg+40),x+2,bf
LINE(5,interv+40)-(5+ok(x),interv+larg+40),1,b
interv=interv+larg+7
NEXT x
'stampa coordinate
LINE(5,200)-(295,200),1 'linea orizzontale
LINE(5,200)-(5,40),1 'linea verticale
GOSUB stampa.titolo
carattere$="topaz.font":size=8:GOSUB carica.font
FOR x=0 TO INT((n-1)/2)
LOCATE 27,(x*5)+1:COLOR 1,x+2:PRINT label$(x)
NEXT x
x1=1
FOR x=INT(((n-1)/2)+1) TO n-1
LOCATE 28,x1:COLOR 1,x+2:PRINT label$(x)
x1=x1+5
NEXT x:COLOR 1,0
jz=LEN(STR$(max))
LOCATE 27,38-jz:PRINT max
RETURN

```

```

piu.meno:
MENU OFF
IF dati<>1 THEN RETURN
CLS
tipo=3
FOR x=0 TO n-1
ok(x)=dato(x)
NEXT x
GOSUB calcola.massimo
'calcola ampiezza massima di ogni barra
larg=290/n-10
'calcola ampiezza massima di ogni barra
FOR x=0 TO n-1
IF ok(x)>0 THEN ok(x)=(ok(x)/max)*75
IF ok(x)<0 THEN ok(x)=-((ok(x)/-max)*75)
NEXT x
'10=intervallo tra una barra e l'altra - 200=base del grafico
interv=10
FOR x=0 TO n-1
LINE(interv,125)-(interv+larg,125-ok(x)),x+2,bf
LINE(interv,125)-(interv+larg,125-ok(x)),1,b
interv=interv+larg+10
NEXT x
'stampa coordinate
LINE(5,125)-(295,125),1 'linea orizzontale
LINE(5,200)-(5,40),1 'linea verticale
GOSUB stampa.titolo
carattere$="topaz.font":size=8:GOSUB carica.font
FOR x=0 TO INT((n-1)/2)
LOCATE 27,(x*5)+1:COLOR 1,x+2:PRINT label$(x)

```

## MENU GRIGLIA

ORIZZONTALE. Consente di ottenere una grigliatura orizzontale (vedere tabella 4).

VERTICALE. Stessa cosa, ma verticale (vedere tabella 4).

ELIMINA. Elimina qualsiasi tipo di griglia in quel momento in uso.

## MENU PALETTE

COLORI. Visualizza qualsiasi immagine a colori ridefinendo completamente la palette.

BIANCO E NERO. Viceversa mostra le immagini in modo monocromatico.

Questa opzione è stata inserita per due motivi: per dare un miglior contrasto a coloro che utilizzano monitor monocromatici (o che non hanno ancora il modulatore per il televisore) e per quelli che non hanno stampanti a colori. In questo caso, comunque, faccio presente che da prove fatte i grafici stampati

```

NEXT x
x1=1
FOR x=INT(((n-1)/2)+1) TO n-1
  LOCATE 26,x1:COLOR 1,x+2:PRINT label$(x)
  x1=x1+5
NEXT x:COLOR 1,0
LOCATE 5,1:PRINT max
RETURN
END
-----
stampa.titolo:
carattere$="ruby.font":size=12:GOSUB carica.font
orizz=(24-LEN(titolo$))/2
LOCATE 2,orizz:PRINT titolo$
RETURN

carica.font:
font1$=carattere$+CHR$(0)
attributo(0)=SADD(font1$)
attributo(1)=size*65536
nomefont=OpenDiskFont(VARPTR(attributo(0)))
CALL SetFont(WINDOW(8).nomefont)
RETURN

calcola.massimo:
max=0:min=0
FOR x=0 TO n-1
  IF ok(x)>max THEN max=ok(x)
  IF ok(x)<min THEN min=ok(x)
NEXT x
IF ABS(min)>max THEN max=ABS(min)
RETURN

SUB stampa STATIC -----
ScreenDump:
sWindow& = WINDOW(7)
sScreen& = PEEKL(sWindow& + 46)
sViewPort& = sScreen& + 44
sRastPort& = sScreen& + 84
sColorMap& = PEEKL(sViewPort& + 4)

maxWidth% = PEEKW(sScreen& + 12)
maxHeight% = PEEKW(sScreen& + 14)
viewModes% = PEEKW(sViewPort& + 32)

command% = 11
srcX% = 0
srcY% = 0
srcWidth% = maxWidth%
srcHeight% = maxHeight%
destRows& = 0
destCols& = 0
special% = &H84

IF BorderFlag% = 0 THEN
  srcX% = srcX% + 3
  srcY% = srcY% + 11
  srcWidth% = srcWidth% - 3 - 11
  srcHeight% = srcHeight% - 11 - 3
END IF

'LIBRARY "exec.library"

sigBit% = AllocSignal%(-1)
ClearPublic& = 655376
msgPort& = AllocMem&(40,ClearPublic&)
IF msgPort& = 0 THEN
  PRINT "Can't allocate msgPort"
  GOTO cleanup4
END IF

POKE(msgPort& + 8), 4 'Type=NT_MSGPORT
POKE(msgPort& + 9), 0 'Priority 0
portName$ = "MyPrtPort"+CHR$(0)
POKEL(msgPort& + 10), SADD(portName$)
POKE(msgPort& + 14), 0 'Flags
POKE(msgPort& + 15), sigBit%
sigTask& = FindTask&(0)
POKEL(msgPort& + 16), sigTask&

CALL AddPort(msgPort&)

ioRequest& = AllocMem&(64,ClearPublic&)
IF ioRequest& = 0 THEN
  PRINT "Can't allocate ioRequest"
  GOTO cleanup3
END IF

POKE(ioRequest& + 8), 5
POKE(ioRequest& + 9), 0
POKEL(ioRequest& + 14), msgPort&

devName$ = "printer.device"+CHR$(0)
pError& = OpenDevice&(SADD(devName$),0,ioRequest&,0)
IF pError& <> 0 THEN
  PRINT "NON TROVO LA STAMPANTE!!!"
  GOTO cleanup2
END IF

POKEW(ioRequest& + 28), command%
POKEL(ioRequest& + 32), sRastPort&
POKEL(ioRequest& + 36), sColorMap&
POKEL(ioRequest& + 40), viewModes%
POKEW(ioRequest& + 44), srcX%
POKEW(ioRequest& + 46), srcY%
POKEW(ioRequest& + 48), srcWidth%
POKEW(ioRequest& + 50), srcHeight%
POKEL(ioRequest& + 52), destCols&
POKEL(ioRequest& + 56), destRows&
POKEW(ioRequest& + 60), special%

ioError& = DoIO&(ioRequest&)
IF ioError& <> 0 THEN
  PRINT "DumpRPort error =" ioError&
  GOTO cleanup1
END IF

cleanup1:
CALL CloseDevice(ioRequest&)

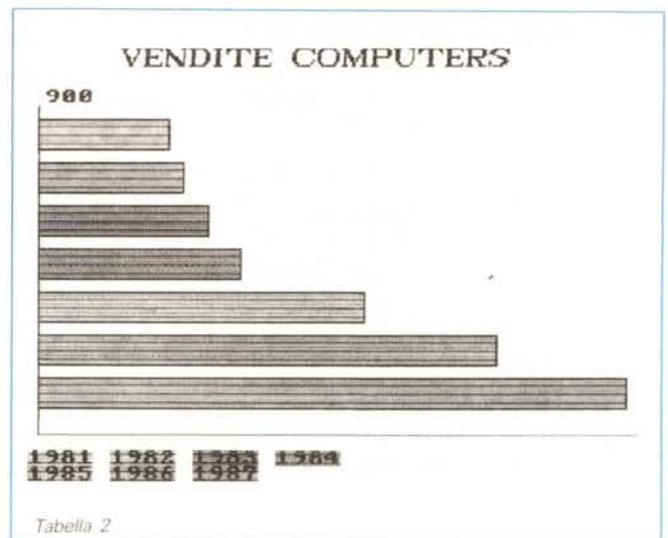
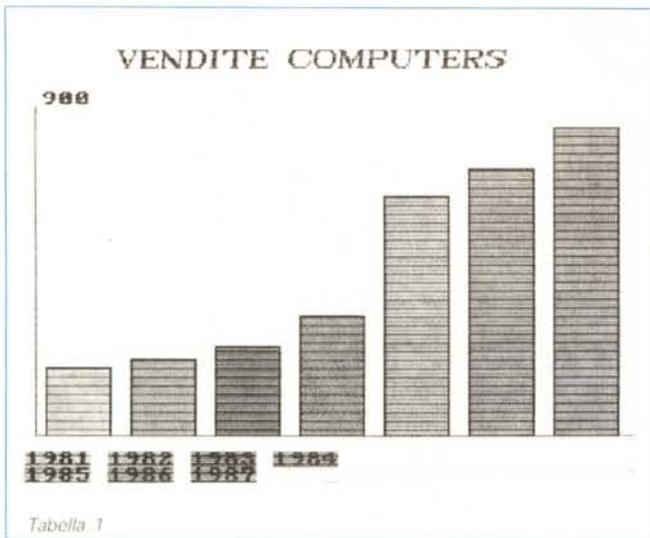
cleanup2:
POKE(ioRequest& + 8), &HFF
POKEL(ioRequest& + 20), -1
POKEL(ioRequest& + 24), -1
CALL FreeMem(ioRequest&,64)

cleanup3:
CALL RemPort(msgPort&)
POKE(msgPort& + 8), &HFF
POKEL(msgPort& + 20), -1
CALL FreeSignal(sigBit%)
CALL FreeMem(msgPort&,40)

cleanup4:
'library close

END SUB -----

```



con stampanti in B&N, ottenuti con la visualizzazione su schermo a colori, risultano altrettanto leggibili e con un buon contrasto cromatico.

#### MENU TIPO

**VERTICALE.** Mostra gli istogrammi verticali dei dati inseriti.

Se è stato inserito un valore negativo, ciò sarà rappresentato come zero e quindi non verrà visualizzato (vedere tabella 1).

**ORIZZONTALE.** Visualizza gli istogrammi orizzontali. Per il resto è uguale al precedente (vedere tabella 2).

**PIU E MENO.** Mostra gli istogrammi verticali, ma se ci sono valori negativi rappresenta anche questi (vedere tabella 3).

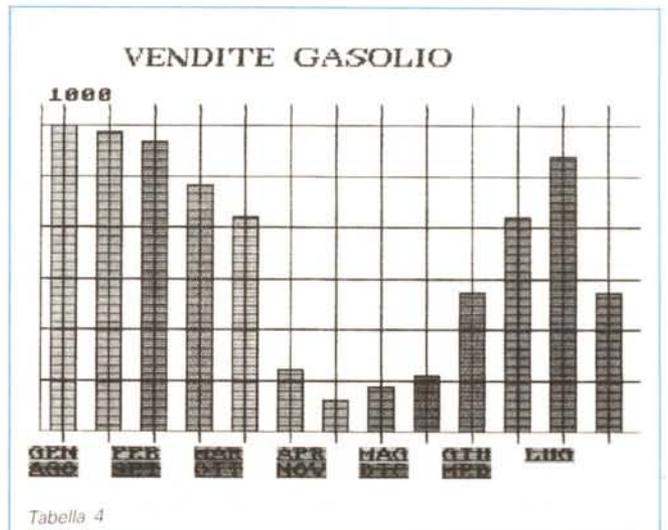
#### Informazioni importantissime

Il programma, per poter funzionare, necessita della presenza dei tipi di carattere RUBY e TOPAZ e delle librerie EXEC, GRAPHICS e DISKFONT. Per quanto riguarda i caratteri non ci sono problemi, dato che si tratta semplicemente di trasferirli dal dischetto Workbench ad Extras forniti a corredo della macchina (nella directory FONTS). Le librerie EXEC e GRAPHICS si trovano nel dischetto Extras e si tratta di copiarle pari pari sul dischetto, che ospita il programma. La libreria DISKFONT non esiste e bisognerà crearla. Ricordo che il suo impiego in questo programma è molto importante dato che si incarica di visualizzare i caratteri pre-

senti nella directory FONTS nelle varie dimensioni (sullo stile del Notepad, tanto per intenderci).

Penso che occorra spendere qualche parola per spiegarne la sua creazione... Inserire l'Workbench; a caricamento avvenuto inserire il dischetto Extras; aprire il «cassetto» BASICDEMOS e caricare il file ConvertFD. Alla richiesta del file in input digitare DISKFONT ed a quella del file in output DISKFONT.BMAP.

Trascorsi alcuni istanti, si avrà quest'ultimo file presente sul disco. Copiarlo sul disco che ospita il programma GRAFICI ed il gioco è fatto. Ovviamente tutto ciò non sarà necessario per chi acquisterà il disco presso la redazione.



TUTTI I MATERIALI  
SONO GARANTITI  
1 ANNO

ECCEZIONALE  
SUPER EGA  
800 x 600  
MONITOR MULTI  
SYNC COLORE  
L. 1.399.000

DISK DRIVE  
360 Kbytes  
L. 157.000

SUPER EGA  
800 x 600  
L. 399.000

WITTY MOUSE  
3 TASTI  
L. 85.000

ACCELERATORE  
SPEED-CARD 286  
L. 392.000

SCHEDA COLOR  
O HERCULES  
L. 99.000

MODEM CARD  
300/1200 HAYES  
L. 255.000

MULTI I/O  
CARD  
L. 160.000

DISK DRIVE  
3.5" - 720 KB  
L. 294.000

PC LOOK XT  
8088 4.77/8MHZ  
256K RAM ESP. 640K  
1 DISK DRIVE 360K  
HERCULES O COLOR  
PORTA PARALLELA  
L. 919.000

CPU  
SYSTEM

PC LOOK AT  
8088 4.77/8MHZ  
256K RAM ESP. 640K  
1 DISK DRIVE 360K  
HARD DISK 20MB  
HERCULES O COLOR  
PORTA PARALLELA  
L. 1.661.000

PC-AT 286  
TURBO 6-12 MHZ  
512K RAM  
1 DISK DRIVE 1.2MB  
1 H.D. 20MB  
HERCULES O COLOR  
PORTA PARALLELA  
PORTA SERIALE  
L. 3.017.000

I PREZZI SUINDICATI SONO IVA ESCLUSA

CPU - 50127 FIRENZE - Via M. Ulivelli 39/r - Tel. 055-4361096 - TELEX 574354 SEAC I - FAX 055/4361096

CPU - 50047 PRATO (FI) - Via Settesoldi 32 - Tel. 0574/434554