

di Francesco Petroni

Finestre sul video in Basic con PC IBM

Le nuove generazioni di software integrato, realizzato specificamente per le macchine 16 bit, fanno largo uso del sistema di visualizzazione per finestre, consistente nel creare più output di formato e contenuto variabile, visualizzabili contemporaneamente sul monitor con una gerarchia che permette di vedere il documento "che sta sopra" e nello stesso tempo di controllare parte dei documenti sottostanti.

Viene così a cadere un'altra caratteristica storica del microcomputer e cioè la rigidità del formato di uscita alfanumerica.

Fino a cinque anni fa si disquisiva sulla convenienza di utilizzare un formato di 16 righe per 64 colonne del TRS 80, un formato di 24 righe per 40 dell'Apple II, o un formato 24 per 80 tipico dei terminali dei mainframe e derivante dal formato delle vecchie schede meccanografiche che così venivano sostituite teoricamente da una riga di monitor.

Il formato 24 per 80 è quindi più professionale ma è sicuramente esuberante in caso di maschere per l'input di pochi dati o insufficiente in caso di lavoro su gran mole di

dati, come ad esempio nell'uso di un foglio elettronico.

Un altro elemento di disagio si ha quando si considera anche l'uscita su stampante che a sua volta può avere i più svariati formati.

E quindi anche i più costosi sistemi di videoscrittura ricorrono a trucchi, come lo scroll laterale e la numerazione delle colonne, per uniformare l'uscita su monitor con quella della stampante. Con la diffusione dei monitor grafici e con il loro uso anche nei soft di base oltre che in quello applicativo il vincolo del formato non è più così rigido; si lavora con le finestre, ognuna delle quali lavora con un proprio formato e un proprio contenuto e ognuna delle quali si può aprire e chiudere a piacimento.

La funzione finestra è ad esempio presente nel PCOS Basic dell'Olivetti M20 ed è anche molto potente in quanto con comodi comandi di linguaggio genera le operazioni di apertura/chiusura, dimensionamento e di uso di questa o quella finestra.

La gestione delle finestre va quindi fatta a livello di sistema operativo o a livello di linguaggio interprete. La gestione tramite subroutine in Basic è possibile, ma a causa della sua lentezza può essere limitata a poche finestre che lavorino contemporaneamente.

Cosa si può fare in Basic

L'argomento che trattiamo è proprio questo; vedremo, usando un IBM PC, come da Basic si possono usare, creare fine-

```

100 REM CinquePagine
110 CLS : REM Main
120 READ X1,Y1,X,Y,A:X2=X1+X:Y2=Y1+Y
130 IF X1=0 THEN END
140 GOSUB 200:GOSUB 300:GOTO 120
200 REM Costruzione Cornice
210 FOR X=X1+1 TO X2-1:LOCATE Y1,X: PRINT CHR$(196):NEXT X
220 FOR X=X1+1 TO X2-1:LOCATE Y2,X: PRINT CHR$(196):NEXT X
230 FOR Y=Y1+1 TO Y2-1:LOCATE Y,X1: PRINT CHR$(179):NEXT Y
240 FOR Y=Y1+1 TO Y2-1:LOCATE Y,X2: PRINT CHR$(179):NEXT Y
250 LOCATE Y1,X1:PRINT CHR$(218):LOCATE Y2,X2:PRINT CHR$(217)
260 LOCATE Y2,X1:PRINT CHR$(192):LOCATE Y1,X2:PRINT CHR$(191):RETURN
300 REM Scrittura
310 FOR X=X1+1 TO X2-1:FOR Y=Y1+1 TO Y2-1:LOCATE Y,X:PRINT CHR$(A)
320 NEXT Y:NEXT X:RETURN
400 REM
410 DATA 1,2,40,13,49
420 DATA 10,4,40,18,50
430 DATA 30,6,30,17,51
440 DATA 50,1,30,20,52
450 DATA 22,8,40,11,53
460 DATA 0,0,0,0,0
  
```

Figura 1 - CinquePagine - Listato del programma. Il programma dimostra la leggibilità di cinque pagine di vario formato, visualizzabili con una certa gerarchia sul video.

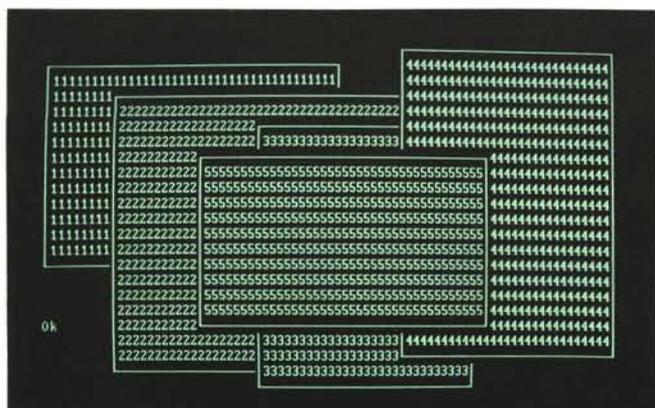


Figura 2 - CinquePagine - Output su video. E opportuno, per migliorare la leggibilità, tracciare delle cornici attorno a ciascuna finestra.

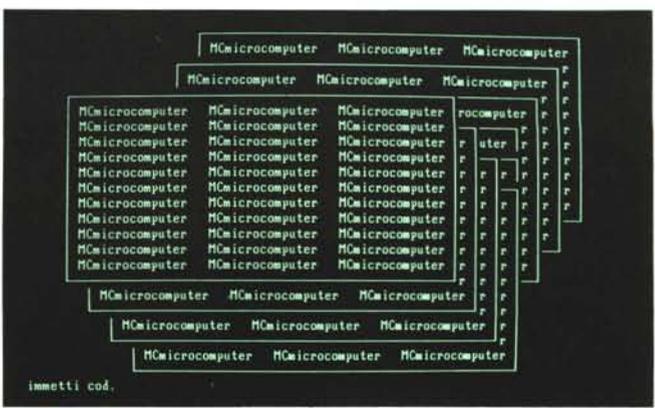


Figura 3 - Sposta - Output su video. Una volta disegnata la nuova cornice è bene cancellare l'interno (con una stringa di blank opportunamente lunga), in quanto il suo contenuto precedente può "sporcare" eventuali spazi vuoti della nuova pagina.

```

100 REM creazione di una finestra
110 CLS: X1=25: Y1=1: X0=54: Y0=12: B$=STRING$(80, 32)
120 A$="MCmicrocomputer": X2=X1+X0: Y2=Y1+Y0: GOSUB 300
130 REM spostamento
140 LOCATE 24,1: PRINT " immetti cod. ";
150 I$=INKEY$: IF I$="" THEN 150
160 ON VAL(I$) GOSUB 210, 220, 230, 240, 150, 250, 260, 270, 280
170 GOTO 150
200 REM Parametri Spostamenti
210 Y1=Y1+2: X1=X1-3: GOSUB 300: RETURN
220 Y1=Y1+2: GOSUB 300: RETURN
230 Y1=Y1+2: X1=X1+3: GOSUB 300: RETURN
240 X1=X1-3: GOSUB 300: RETURN
250 X1=X1+3: GOSUB 300: RETURN
260 Y1=Y1-2: X1=X1-3: GOSUB 300: RETURN
270 Y1=Y1-2: GOSUB 300: RETURN
280 Y1=Y1-2: X1=X1+3: GOSUB 300: RETURN
300 REM cornice
310 X2=X1+X0: Y2=Y1+Y0
320 FOR X=X1+1 TO X2-1: LOCATE Y1, X: PRINT CHR$(196): NEXT X
330 FOR X=X1+1 TO X2-1: LOCATE Y2, X: PRINT CHR$(196): NEXT X
340 FOR Y=Y1+1 TO Y2-1: LOCATE Y, X1: PRINT CHR$(179): NEXT Y
350 FOR Y=Y1+1 TO Y2-1: LOCATE Y, X2: PRINT CHR$(179): NEXT Y
360 LOCATE Y1, X1: PRINT CHR$(218): LOCATE Y2, X2: PRINT CHR$(217)
370 LOCATE Y2, X1: PRINT CHR$(192): LOCATE Y1, X2: PRINT CHR$(191)
380 REM cancellazione
390 FOR Y=Y1+1 TO Y2-1: LOCATE Y, X1+1: PRINT LEFT$(B$, X0-1): NEXT Y
400 REM scritte
410 FOR Y=Y1+1 TO Y2-1: FOR X=X1+2 TO X2 STEP 18
420 LOCATE Y, X: PRINT A$: NEXT X: NEXT Y: RETURN

```

Figura 4 - Sposta - Listato del programma. Visualizzata una prima volta la finestra, la si può spostare in giro per il video utilizzando il tastierino numerico.

```

100 REM PAGINA 1
110 CLS: C$=STRING$(3, 178): D$=STRING$(47, 196): E$=STRING$(47, 255)
120 FOR I=1 TO 8: READ C: C$(I)=CHR$(C): NEXT I
130 LOCATE 2,6: PRINT " soc. XYW ANDAMENTO VENDITE "
140 FOR I=70 TO 0 STEP -10: LOCATE 18-2*I/10, 1: PRINT I: NEXT I
150 FOR I=5 TO 52: LOCATE 18, I+1: IF I/4 <> INT(I/4) THEN PRINT C$(2) ELSE
PRINT C$(1): READ M$: LOCATE 19, I-1: PRINT M$
160 NEXT I: LOCATE 20, 6: PRINT " ANNO 1984 ": FOR I=3 TO 17: LOCATE I, 5
170 IF I/2 = INT(I/2) THEN PRINT C$(3)+D$+C$(6): ELSE PRINT C$(4)+E$+C$(4)
180 NEXT I: LOCATE 18, 5: PRINT C$(5): LOCATE 18, 53: PRINT C$(8);
190 FOR I=9 TO 52 STEP 4: FOR L=3 TO 17: LOCATE L, I
200 IF L/2 = INT(L/2) THEN PRINT C$(7): ELSE PRINT C$(4)
210 NEXT L: NEXT I: FOR I=1 TO 12: READ D: L=I*4+2: FOR K=0 TO D
220 LOCATE 17-K, L: PRINT C$: NEXT K: NEXT I: GOTO 260
230 DATA 193, 196, 195, 179, 192, 180, 197, 217
240 DATA gen, feb, mar, apr, mag, giu, lug, ago, set, ott, nov, dic
250 DATA 8, 9, 11, 12, 10, 12, 10, 9, 11, 12, 13, 11
260 REM cornice
270 X1=1: Y1=1: X0=54: Y0=20: X2=X1+X0: Y2=Y1+Y0: DIM SC$(Y0, X0)
280 FOR X=X1+1 TO X2-1: LOCATE Y1, X: PRINT CHR$(205): NEXT X
290 FOR X=X1+1 TO X2-1: LOCATE Y2, X: PRINT CHR$(205): NEXT X
300 FOR Y=Y1+1 TO Y2-1: LOCATE Y, X1: PRINT CHR$(186): NEXT Y
310 FOR Y=Y1+1 TO Y2-1: LOCATE Y, X2: PRINT CHR$(186): NEXT Y
320 LOCATE Y1, X1: PRINT CHR$(201): LOCATE Y2, X2: PRINT CHR$(188)
330 LOCATE Y2, X1: PRINT CHR$(200): LOCATE Y1, X2: PRINT CHR$(187)
340 REM immagazzinamento
350 FOR Y=Y1 TO Y2: FOR X=X1 TO X2: SC$(Y-Y1, X-X1)=SCREEN(Y, X): NEXT X, Y
360 REM visualizzazione
370 X3=25: Y3=3: FOR Y=0 TO Y0: FOR X=0 TO X0: LOCATE Y+Y3, X+X3
380 PRINT CHR$(SC$(Y, X)): NEXT X: NEXT Y
390 I$=INKEY$: IF I$="" THEN 390
400 REM creazione di un file sequenziale
410 OPEN "PAG1.DAT" FOR OUTPUT AS #1: WRITE #1, Y0, X0
420 FOR Y=0 TO Y0: FOR X=0 TO X0
430 WRITE #1, SC$(Y, X): NEXT X: NEXT Y: CLOSE #1: END

```

Figura 5 - Pagina 1 - Listato del programma. Solito istogramma che viene immagazzinato come insieme di caratteri ASCII in una matrice di numeri interi.

stre, memorizzarle, spostarle sul video, combinarle con altre, ecc.

Come noto il Basic è estremamente potente nella gestione dei dati sotto forma di stringa e quindi le modalità di manipolazione dei dati da visualizzare e/o già visualizzati sono molteplici.

Le istruzioni di cui l'advanced Basic dispone e che possono essere utilizzate nei vari problemi che ci si presentano sono:

— tutte le funzioni di stringa

— la funzione di posizionamento su video LOCATE Y,X

— la funzione di lettura da una posizione sul video A = SCREEN (Y,X) che cioè, legge il codice ASCII presente in quella posizione del video

— A = PEEK (LOC) che fa la stessa cosa in quanto legge il codice ASCII del carattere presente nella cella LOC della memoria. Se LOC è una locazione interna alla video memory l'A corrisponde al carattere visualizzato.

Inoltre il BASIC del PC comprende un set di 256 caratteri, di cui molti grafici, ciascuno dei quali è visualizzabile in 11 modi differenti; per leggere anche la modalità di visualizzazione dal video occorre quindi leggere due dati, carattere visualizzato e modalità di visualizzazione. In fase di scrittura si risparmia in quanto la modalità di visualizzazione se è comune a più caratteri può essere fornita una sola volta (si pensi alla riga titolo, scritta con carattere intenso).

Anche per memorizzare il contenuto delle varie finestre si possono seguire varie strade. Quella di memorizzare i dati in matrici numeriche rettangolari il cui elemento rappresenta il codice del carattere ASCII corrispondente, o di memorizzare direttamente il carattere come stringa, oppure si può rendere stringa un'intera riga della finestra o se del caso più righe, ecc.

In questo ultimo caso si ha la massima velocità di scrittura e la massima economia della memorizzazione.

Le possibilità sono tante e vanno scelte in funzione degli scopi precisi che si vogliono raggiungere.

Utilizzando il PC IBM, ma questo vale anche per gli altri microcomputer che hanno set di caratteri comprendenti caratteri grafici, si può trattare come stringa un qualsiasi insieme di caratteri anche grafici e quindi una intera videata, ad esempio, può essere memorizzata in un vettore V\$(25) in cui il singolo elemento V\$(I) è una stringa di 80 caratteri.

In questa maniera si rinuncia a memorizzare la modalità di visualizzazione che

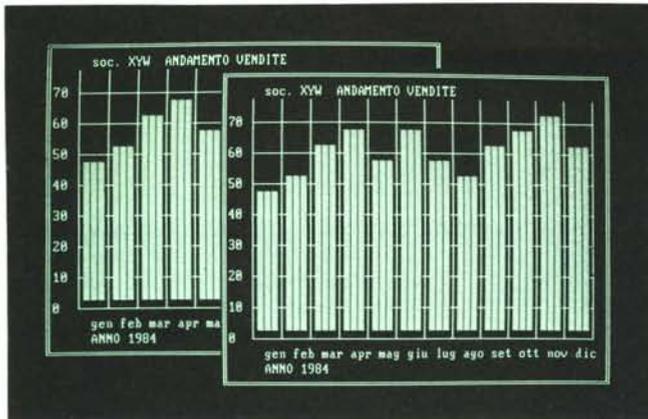


Figura 6 - Pagina 1 - Output su video. La matrice di interi (codici ASCII) viene tradotta di nuovo nei corrispondenti caratteri, in un'altra posizione sul video.

può essere memorizzata a parte con altri sistemi.

Va inoltre detto che nessuna delle modalità di visualizzazione viene riportata su carta facendo l'hard copy del video su printer (inverso, lampeggiante, sottolineato, intenso, ecc.) e qualcuna non può essere neanche riprodotta in fotografia (ad esempio il lampeggio).

I programmi

Passiamo ad esaminare i programmi dimostrativi a corredo dell'articolo. Sono tutti realizzati con un PC IBM, utilizzato in modo alfanumerico.

Anche questa volta il discorso è generalizzabile e trasferibile su qualsiasi altro computer, si tratta di utilizzare altri set di caratteri, altre istruzioni di localizzazione e di visualizzazione, altri formati video, ma tutte le problematiche e le soluzioni tecniche sono facilmente trasferibili.

Programma cinquepagine

Il programma serve per valutare l'efficacia di un sistema a finestre e la leggibilità di ciascuna di esse.

Innanzitutto è indispensabile utilizzare delle cornici per ciascuna finestra; queste possono essere realizzate con gli opportuni

caratteri grafici. Il problema della sovrapposizione tra le cornici potrebbe essere risolto con una subroutine che utilizzi altri caratteri speciali. Ma il gioco non vale la candela in quanto la routine di riconoscimento sarebbe molto più pesante a fronte della visualizzazione di qualche carattere di cui non si sente assolutamente la mancanza.

Il programma ha una MAIN routine che legge i DATA e richiama le due subroutine.

I DATA sono:

X1, Y1 coordinate del vertice in alto a sinistra della finestra,

X, Y larghezza ed altezza della finestra, A carattere da visualizzare nella specifica finestra.

Utilizzeremo anche negli altri programmi questo sistema di costruzione delle finestre.

Le variabili X2, Y2 sono le coordinate, calcolate dal programma, del vertice in basso a sinistra.

Programma sposta

Questo programma utilizza un'unica finestra contenente un certo messaggio, che viene spostata in giro per il video semplicemente utilizzando le otto direzioni del ta-

stierino numerico. Dopo le inizializzazioni (righe 100 ÷ 120) c'è la routine che gestisce lo spostamento delle finestre (righe 130 ÷ 280).

Premendo un tasto del tastierino numerico la finestra si sposta nella direzione voluta cancellando quello che viene coperto dalla nuova posizione e lasciando scoperto il resto.

La cancellazione è la routine di riga 380 e la scrittura è quella di riga 400.

Ovviamente la routine di riempimento può essere sostituita con una qualsiasi delle altre viste negli altri programmi.

Programma pagina 1

Il secondo programma si svolge in quattro fasi successive.

La prima fase consiste nella creazione di un istogramma (da riga 100 a riga 250) in un modo ripreso da programmi pubblicati nei numeri scorsi. Attorno a questo istogramma viene tracciata una cornice (da riga 260 a riga 330) utilizzando alcuni caratteri grafici.

Il tutto viene memorizzato in una matrice di interi di dimensioni pari a quella della finestra, prelevando il codice ASCII di ciascun carattere visualizzato.

La terza fase è la visualizzazione carattere per carattere in una posizione differente. In sostanza i parametri passati alle istruzioni SCREEN (Y,X) e LOCATE (Y,X) differiscono di fattori di spostamento X3,Y3. Questo sistema è lento in quanto lavora carattere per carattere sia in lettura che in scrittura ed inoltre usa per ciascun carattere i due byte che servono per un numero intero.

L'ultima fase consiste nella memorizzazione in un file sequenziale (chiamato PAG. 1. DAT) delle dimensioni della matrice e del suo contenuto, in modo da permettere la lettura e la visualizzazione anche da altri programmi.

Programma misto

Scopo del programma è quello di visualizzare più pagine realizzate ciascuna con una modalità differente.



Figura 8 - Pagina 2 - Output su video. In fase di out le stringhe memorizzate vanno posizionate opportunamente sullo schermo, con la istruzione LOCATE.

```

100 REM pagina 2
110 CLS:A=1975:X1=1:Y1=1:X0=54:Y0=13:X2=X1+X0:Y2=Y1+Y0:DIM A$(Y0+1)
120 FOR X=X1+1 TO X2-1:LOCATE Y1,X:PRINT CHR$(196):NEXT X
130 FOR X=X1+1 TO X2-1:LOCATE Y2,X:PRINT CHR$(196):NEXT X
140 FOR Y=Y1+1 TO Y2-1:LOCATE Y,X1:PRINT CHR$(179):NEXT Y
150 FOR Y=Y1+1 TO Y2-1:LOCATE Y,X2:PRINT CHR$(179):NEXT Y
160 LOCATE Y1,X1:PRINT CHR$(218):LOCATE Y2,X2:PRINT CHR$(217)
170 LOCATE Y2,X1:PRINT CHR$(192):LOCATE Y1,X2:PRINT CHR$(191)
180 LOCATE 2,8:COLOR 0,7:PRINT "Produzione Vino Rosso ( Mil. di HL )"
190 COLOR 7,0:FOR I=1 TO 8:II=I+3:LOCATE II,2:PRINT A+I:LOCATE II,8
200 P=INT(40+RND(1)*40):PP=PP+P:FOR L=1 TO P/2:PRINT CHR$(219):NEXT L
210 IF P/2 = INT(P/2) THEN PRINT CHR$(221);
220 NEXT I:LOCATE 13,8:PRINT "Produzione Media ";USING "###.###";PP/8
230 FOR V=1 TO 14:FOR O=1 TO 55:A$(V)=A$(V)+CHR$(SCREEN(V,O))
240 NEXT O:NEXT V
250 REM visualizzazione
260 CLS:FOR V=1 TO 14:PRINT A$(V):NEXT V
270 REM creazione di un file sequenziale
280 OPEN "PAG2.DAT" FOR OUTPUT AS #1
290 FOR V=1 TO 14:WRITE #1,A$(V):NEXT V:CLOSE #1:END

```

Figura 7 - Pagina 2 - Listato del programma. In questo caso la memorizzazione avviene per stringhe (una per riga) che quindi vengono visualizzate molto velocemente.

```

100 REM Misto Pagine
110 NT=2:B#=STRING$(80,32):L#=STRING$(80,196):CLS:FOR I=1 TO NT
120 READ NP,FX,FY,X0,Y0,T$:GOSUB 140:FOR L=1 TO NR
130 READ R$:GOSUB 170:NEXT L:GOSUB 190:NEXT I:GOTO 210
140 REM scrittura intestazione
150 LT=FX-8:Y2=Y0+FY-1:X2=X0+FX:NR=FY-2:LR=FX-1
160 LOCATE Y0,X0:COLOR 0,7:PRINT CHR$(221);LEFT$( " "+T$+B$,LT);"Pag. ";NP
CHR$(222);:COLOR 7,0:RETURN
170 REM scrittura riga
180 LOCATE Y0+L,X0:PRINT CHR$(179);LEFT$(R$+B$,LR);CHR$(179):RETURN
190 REM scrittura ultima riga
200 LOCATE Y2,X0:PRINT CHR$(192);LEFT$(L$,LR);CHR$(217):RETURN
210 REM programma di lettura e visualizzazione files seq.
220 GOSUB 300:OPEN "PAG1.DAT" FOR INPUT AS #1
230 INPUT #1,Y0,X0:FOR Y=0 TO Y0:FOR X=0 TO X0:INPUT #1,C%
240 LOCATE Y+3,X+25:PRINT CHR$(C%):NEXT X:NEXT Y:CLOSE #1
250 GOSUB 300:OPEN "PAG2.DAT" FOR INPUT AS #1
260 FOR V=1 TO 14:INPUT #1,A$:LOCATE 1+V,6:PRINT A$:NEXT V:CLOSE #1
270 GOSUB 300:OPEN "PAG3.DAT" FOR INPUT AS #1
280 INPUT #1,A$:LOCATE 12,1:PRINT A$:CLOSE #1
290 I$=INKEY$:IF I$="" THEN 290 ELSE CLS:END
300 I$=INKEY$:IF I$="" THEN 300 ELSE RETURN
310 REM data prime due pagine
320 DATA 1,46,7,20,1,VENDITA AUTO
330 DATA " Carissimi Lettori
340 DATA " vi comunichiamo che la vendita delle auto-
350 DATA " mobili ha subito una lieve flessione dovuta
360 DATA " alla politica comunitaria nei confronti del-
370 DATA " la produzione giapponese
380 DATA 2,50,10,1,9,Prossime Elezioni
390 DATA " le notizie riferite alla agenzia di stampa
400 DATA " Nuova Cina risultano a detta del portavoce
410 DATA " ufficiale del governo destituite di ogni
420 DATA " fondamento.
430 DATA " E' per questo motivo e non per altri che il
440 DATA " Presidente del Consiglio dei Ministri ieri
450 DATA " ha ricevuto a Palazzo Chigi l' Ambasciatore
460 DATA " cinese TSIEN LION PUNG

```

Figura 9 - Misto - Listato del Programma. Il sistema più semplice per memorizzare il contenuto di una finestra è quello di utilizzare i DATA.

La modalità più semplice è quella di inserire caratteristiche e contenuto di una finestra direttamente all'interno del programma con le istruzioni DATA.

Le caratteristiche, che vengono ovviamente lette da apposite istruzioni di READ, sono:

NP numero della pagina;
FX, FY dimensioni della finestra;
X0, Y0 posizione del vertice in alto a sinistra;
T\$ titolo della finestra;

Il contenuto viene letto con un loop di lunghezza pari al numero delle righe, che così vengono lette una per una.

La visualizzazione di ciascuna finestra così immagazzinata avviene in tre fasi successive:

titolo (righe 140-160),
corpo del testo (righe 170-180)
ultima riga (righe 190-200)

La cornice viene disegnata contestualmente in ciascuna fase.

Oltre a due pagine così realizzate il programma MISTO visualizza tre altre pagine provenienti dai file sequenziali, e abbiamo visto prima in che modo vengono costruiti.

PAG. 1 DAT contiene una finestra co-

struita carattere per carattere (righe 220-240);

PAG. 2 DAT contiene una finestra costruita riga per riga (righe 250-260);

PAG. 3 DAT contiene una finestra costituita da un solo carattere (righe 270-280)

Per far avanzare il programma occorre premere un tasto qualsiasi (riga 290) e così pure per farlo finire (riga 300).

È evidente lo scopo dimostrativo di tale programma in quanto non ha senso "collezionare" un insieme di modi differenti di risolvere lo stesso problema.

Programma pagina 2

Il programma consiste nella realizzazione, eseguita al solito modo, di un istogramma, inserito al solito modo, in una cornice (righe 200-220).

L'utilizzazione degli istogrammi orizzontali comporta, sul video alfanumerico, il vantaggio di una maggiore definizione (fino ad oltre 150 gradini) in quanto rende possibile anche l'uso del mezzo carattere CHR\$(221).

Cioè se il dato da visualizzare ha formato intermedio tra due colonne, con il mezzo carattere si evidenzia tale gradino intermedio.

Nella parte finale (righe 230-240) viene costruito un vettore di stringhe A\$(V) cui corrisponde riga per riga l'istogramma.

Questo viene prima visualizzato di nuovo in maniera veloce, trattandosi di poche stringhe, e poi viene memorizzato nel file sequenziale PAG. 2 DAT come sequenza di stringhe, anche questa volta in vista di una loro utilizzazione successiva.

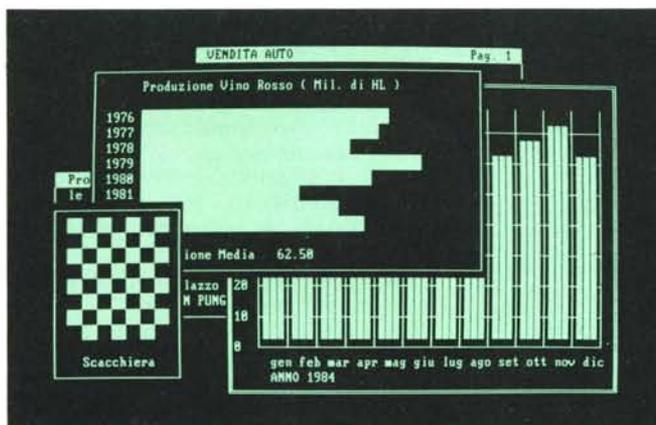


Figura 10 - Misto - Output su video. Delle cinque pagine visualizzate, due provengono da dati inseriti nel programma, le altre sono gli out memorizzati su file sequenziali dei programmi PAG1, PAG2, PAG3 visti in questo stesso articolo.

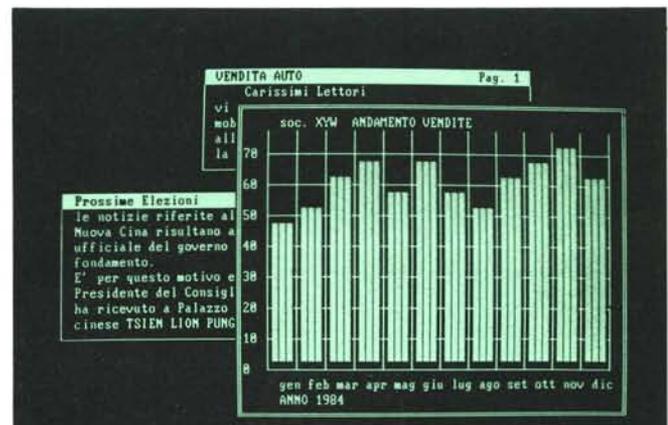


Figura 11 - Misto - Output su video. Dopo la visualizzazione delle prime tre finestre la situazione è questa. Per avanzare basta premere un tasto qualsiasi.

```

100 REM pagina 3
110 A=77:X1=1:Y1=1:X0=17:Y0=11:X2=X1+X0:Y2=Y1+Y0:O$=CHR$(196):V$=CHR$(179)
120 R$=CHR$(255)+CHR$(255)+CHR$(219)+CHR$(219)
130 R1$=CHR$(219)+CHR$(219)+R$+R$+R$:R2$=R$+R$+R$+CHR$(255)+CHR$(255)
140 CLS:FOR X=X1 TO X2:LOCATE Y1,X:PRINT O$:LOCATE Y2,X:PRINT V$:NEXT X
150 FOR Y=Y1 TO Y2:LOCATE Y,X1:PRINT R$:LOCATE Y,X2:PRINT R$:NEXT Y
160 LOCATE Y1,X1:PRINT CHR$(218):LOCATE Y2,X2:PRINT CHR$(217)
170 LOCATE Y2,X1:PRINT CHR$(192):LOCATE Y1,X2:PRINT CHR$(191)
180 FOR V=2 TO 9 STEP 2:LOCATE V,3:PRINT R1$
190 LOCATE V+1,3:PRINT R2$:NEXT V:LOCATE 11,5:PRINT "Scacchiera"
200 FOR V=1 TO 12:FOR O=1 TO 18:A$=A$+CHR$(SCREEN(V,O)):NEXT O
210 A$=A$+CHR$(13):NEXT V
220 REM Loop di Visualizzazione
230 CLS:FOR I=1 TO 9 STEP 2:LOCATE I,1:PRINT A$:NEXT I
240 REM creazione di un file sequenziale
250 OPEN "PAG3.DAT" FOR OUTPUT AS #1:WRITE #1,A$:CLOSE #1:END

```

Figura 12 - Pagina 3 - Listato del Programma. Il carattere CR (CHR\$(13)) va memorizzato alla fine di ogni riga. L'unico limite alla creazione di una stringa è il numero massimo di caratteri (non più di 254).

Programma pagina 3

Il programma consiste nella creazione di una finestra che rappresenta una scacchiera (righe 100-190) e nella composizione di una stringa A\$, comprendente tutta la finestra (200-210), nella sua ripetuta visualizzazione (220-230) eseguita per vedere se ha funzionato e nella memorizzazione della stringa nel file sequenziale PAG 3. DAT (240-250).

È una esasperazione dei concetti precedenti. Si ha la massima economia di

memoria e la massima velocità di visualizzazione, ma è ben raro che una intera finestra possa essere contenuta in un'unica stringa.

Gli ultimi due programmi

Non hanno uno specifico significato pratico né alcun aspetto grafico, ma rappresentano delle utility.

Il primo (SUGIU di fig. 14) espone un metodo, abbastanza elementare, per ottenere lo scroll di un testo più lungo del

```

100 REM SU e GIU
110 CLS:P=1:DIM A$(50):B$=STRING$(20,32)
120 FOR R=1 TO 50:PRINT R:FOR C=1 TO 10+INT(RND(1)*10)
130 A$(R)=A$(R)+CHR$(INT(RND(1)*25+65)):NEXT C:NEXT R:CLS:GOSUB 190
140 DEF SEG=0:POKE 1050,PEEK(1052)
150 I$=INKEY$:IF I$="" THEN 150
160 IF I$="d" OR I$="D" THEN P=P+1:GOSUB 190:GOTO 150
170 IF I$="u" OR I$="U" THEN P=P-1:GOSUB 190:GOTO 150
180 CLS:PRINT "Bye":END
190 LOCATE 1,1:FOR R=P TO P+21:PRINT USING "#####";R:PRINT TAB(6)A$(R);B$
200 NEXT R:RETURN

```

Figura 14 - Sugiu - Listato del Programma. Il programma visualizza 22 righe di testo composto da un numero maggiore di righe. Premendo U o D il testo scorre su e giù. Viene gestito tramite un contatore.

```

100 REM finestre PC IBM
110 DEF SEG
120 GOSUB 170:POKE &H5B,5 * Top
130 GOSUB 170:POKE &H5C,10 * Down
140 GOSUB 170:POKE &H29,40 * Right
150 GOSUB 170
160 POKE &H5B,1 : POKE &H5C,25 :POKE &H29,80: END
170 CLS:FOR I=100 TO 500:PRINT I:NEXT I:RETURN:

```

Figura 15 - Finestre Listato del programma. Esempio di utilizzazione dei POKE che gestiscono la window propria del PC IBM e che corrispondono ai famosi POKE 32,.... POKE 33,.... ecc. dell'Apple II.

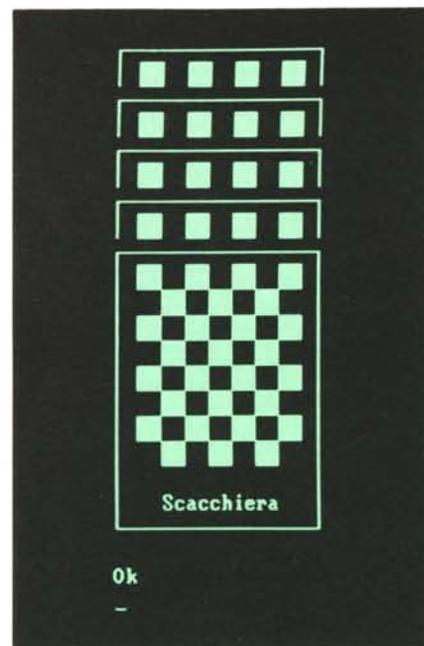


Figura 13 - Pagina 3 - Output su video. Questa è un'unica stringa, contenente ovviamente anche dei caratteri "a capo", e quindi può essere visualizzata solo a sinistra sul video.

formato video, su e giù per il video.

Il testo è costruito in modo random (righe 120-130).

Vengono visualizzate 22 righe per volta; premendo U scorrono in su e premendo D scorrono in giù.

La riga 140 contiene una istruzione necessaria per lo svuotamento del buffer di tastiera da eseguire prima dell'immissione del codice U/D.

Infatti poiché questo avviene via INKEY\$, un'eventuale sporcizia del buffer potrebbe infastidire il programma.

La routine di visualizzazione (righe 190-200) fa vedere le 22 righe indicate dal contatore P.

L'ultimo programma serve per indicare la locazione di memoria che definisce la window ottenibile via SW sul PC IBM. POKE &H005B,Y1 - Y1 identifica il numero di righe da tenere fisse in alto sul video durante lo scroll; POKE &H005C,Y2 - stesso discorso per le righe in fondo alla pagina; POKE &H0029,X - in questo caso rimangono bloccate le colonne a partire dalla X-ma fino alla 80-ma

Il programma gioca un po' con le finestre facendoci scorrere un loop di conteggi (riga 170). Poi, prima di finire mette le cose a posto (riga 160).



Apple IIc, Macintosh Tutti tuoi per un weekend



ROMA - VIA G. LANZA 101 - 103 - 105 - Tel. 738224-738854 **M** VIA VITTORIO EMANUELE (linea A) V. CAVOUR (linea B)
 OSTIA LIDO - VIA ARISTIDE CARABELLI 108 - 110 - 112 - Tel. 5697686 **SABATO APERTO**

CORSI DI BASIC APPLESOFT