

# Listing 1

```

10 INK 7,10:INK 5,26
20 GPAP=0:GPEN=12:GOSUB 110:PRINT" FORCE mode crtanja"
30 PRINT CHR$(23)"0":GOSUB 130
40 GOSUB 110:PRINT" XOR mode crtanja"
50 PRINT CHR$(23)"1":GOSUB 130
60 GPAP=15:GPEN=5:GOSUB 110:PRINT" AND mode crtanja"
70 PRINT CHR$(23)"2":GOSUB 130
80 GPAP=0:GPEN=3:GOSUB 110:PRINT" OR mode crtanja"
90 PRINT CHR$(23)"3":GOSUB 130
100 MODE 1:END
110 MODE 0:CLG GPAP:PLOT -10,-10,GPEN:WINDOW#1,4,9,18,22:WINDOW SWAP 0,1:PAPER 1
:CLS:PEN 0:PRINT" 0001
120 WINDOW#2,12,17,18,22:WINDOW SWAP 0,2:INK 15,4:PAPER 7:CLS:PRINT" 0111":WINDC
W SWAP 0,1:PAPER GPAP:LOCATE 2,12:PRINT"GPAP="BIN$(G
PAP,4)" GPEN="BIN$(GPEN,4):LOCATE 1,1:RETURN
130 FOR N=90 TO 100 STEP 2:MOVE 0,N:DRAWR 639,0:NEXT:CALL &BB06:RETURN

```

# Listing 2

```

10 MODE 2:PRINT CHR$(23)"1":FOR N=0 TO 639:MOVE N/4,0:DRAW N,399:NEXT
20 CALL &BB06:CLG:FOR N=0 TO 639:MOVE 320,399:DRAW N,0:NEXT:CALL &BB06

```

# Listing 3

```

10 MODE 2:INPUT"Unesi koordinate tocaka:",X0,Y0,X1,Y1:Y0=Y0\2:Y1=Y1\2
20 IF X0>X1 THEN Q=X0:X0=X1:X1=Q:Q=Y0:Y0=Y1:Y1=Q
30 XLEN=ABS(X0-X1):YLEN=ABS(Y0-Y1)
40 IF YLEN>XLEN THEN FLAG=2 ELSE FLAG=1 'Da li se mijenja X ili Y varijabla?
50 IF Y0>=Y1 THEN INCDEC=-1 ELSE INCDEC=1
60 IF Y0>=Y1 AND YLEN-XLEN>=0 THEN X0=X1:Y0=Y1
70 XLEN=XLEN+1:YLEN=YLEN+1
80 IF XLEN>YLEN THEN A=XLEN\YLEN:RMAIN=XLEN MOD YLEN:COUNTER=YLEN ELSE A=YLEN\XL
EN:RMAIN=YLEN MOD XLEN:COUNTER=XLEN '\ je znak za in
teger dijeljenje
90 Y8=XLEN\2 'Pocetna vrijednost Y8 varijable
100 FOR N=COUNTER TO 1 STEP -1 'Glavna petlja
110 Y8=Y8+RMAIN:IF Y8>COUNTER THEN LLEN=A+1:Y8=Y8-COUNTER ELSE LLEN=A
120 ON FLAG GOSUB 160,190 'Vodoravno ili okomito?
130 NEXT:CALL &BB06:GOTO 10
140 '
150 'Potprogram za crtanje vodoravnih dijelova linije
160 FOR I=X0 TO X0+LLEN-1:PLOT I,Y0*2,1:NEXT:Y0=Y0+INCDEC:X0=X0+LLEN:RETURN
170 '
180 'Potprogram za crtanje okomitih dijelova linije
190 FOR I=Y0 TO Y0+LLEN-1:PLOT X0,I*2,1:NEXT:X0=X0+INCDEC:Y0=Y0+LLEN:RETURN

```

# Listing 4

```

100 MEMORY HIMEM-13:IF HIMEM<&3FFF THEN PRINT"RAM TOP prenizak!":STOP
110 FOR N=HIMEM+1 TO HIMEM+13:READ A$:POKE N,VAL("&"+LEFT$(A$,2)):NEXT
120 POKE HIMEM+12,PEEK(&BDE9):POKE HIMEM+13,PEEK(&BDEA):NEW
130 H=INT((HIMEM+1)/256):L=((HIMEM+1)/256-H)*256:POKE &BDE9,L:POKE &BDEA,H
140 '
150 DATA E5          PUSH HL
160 DATA 21,00,00    LD HL,0
170 DATA 2B          DO: DEC HL
180 DATA 7C          LD A,H
190 DATA B5          OR L
200 DATA 20,FB       JR NZ,DO
210 DATA E1          POP HL
220 DATA C3,00,00    JP DRAW

```

```

350 ;Linije 10-340 su iste kao i u prošlom broju.
360
370 TABREL: DEFW L2-INIT+1,L3-INIT+1,L3A-INIT+1,L3B-INIT+1,L4-INIT+1
380          DEFW L5-INIT+2,L6-INIT+1,L7-INIT+2
390          DEFW L8-INIT+2,L9-INIT+1,L10-INIT+1,COMTAB-INIT
400          DEFW L11-INIT+1,L12-INIT+1,L13-INIT+1,L14-INIT+1,L14A-INIT+1
410          DEFW DSHTAB-INIT+1,L15-INIT+1,L16-INIT+1,L17-INIT+1,MASK1-INIT+1
420          DEFW MASK2-INIT+1,L18-INIT+1,L18A-INIT+1,L18B-INIT+1
430          DEFW LINE-INIT+1,L19-INIT+1,L20-INIT+1,L21-INIT+1,L22-INIT+1
440          DEFW L23-INIT+1,L25-INIT+1,L26-INIT+1,JP1c-INIT+1,JUMPDR-INIT+2
450          DEFW BCJPSC-INIT+2,L27-INIT+1,L28-INIT+1,LINEex-INIT+1
460          DEFW SCRWR-INIT+1,L29-INIT+1,L30-INIT+1,L31-INIT+1,L32-INIT+1
470          DEFW L33-INIT+1,L34-INIT+1,L35-INIT+1,L36-INIT+1,L37-INIT+1
480          DEFW L38-INIT+1,0
490
500 DONE:   LD     HL, (#BDE3)           ;IND:GRA LINE;Ovo je inicijalizacija.
510 L2:     LD     (JPDR-INIT),HL       ;Prvo treba promijeniti link za
520 L3:     LD     HL,LINE-INIT         ;DRAW (i DRAWR) naredbu.Adresa
530         LD     (#BDE3),HL          ;na koju je trebalo skočiti se čuva.
540         LD     HL, (#BDE9)         ;IND:SCR WRITE
550 L3A:    LD     (JPWRM-INIT),HL      ;Memoriranje SCR WRITE adrese u ROMu.
560         LD     HL, (#BDE6)         ;IND:SCR READ
570 L3B:    LD     (JPSCRRD-INIT),HL    ;Memoriranje SCR READ adrese.
580         CALL #B906                ;KL L ROM ENABLE;Otvaranje donjeg ROMa.
590         LD     HL, (#BBE2)         ;Uzimamo adresu GRA GET PEN rutine.
600 L4:     CALL ONEP-INIT             ;Ta rutina počinje sa LD A,(GPEN)
610 L5:     LD     (PEN+1-INIT),DE      ;pa nam je potrebna apsolutna adresa
620         LD     HL, (#BBE8)         ;iza LD A.Dobivena adresa služi za
630 L6:     CALL ONEP-INIT             ;oformljivanje koda koji se izvršava
640 L7:     LD     (PAP1+1-INIT),DE     ;i na 464 i na 6128.Isto se radi i sa
650 L8:     LD     (PAP2+1-INIT),DE     ;rutinom GRA GET PEN.
660 ;Umjesto LD A,(GPEN) moglo je biti CALL #BBE1:CALL #BC2C (INK ENCODE)
670 ;ali to je čisto gubljenje vremena.
680         CALL #B909                ;KL L ROM DISABLE;Zatvaranje ROMa.
690 L9:     LD     HL,KERSP-INIT       ;Slijedi inicijalizacija RSX
700 L10:    LD     BC,COMTAB-INIT      ;komandi na uobičajeni način.
710         JP     #BCD1               ;KL LOG EXT
720
730 ONEP:   RES    7,H                 ;Ovaj potprogram podatke koji stoje
740         INC    HL                   ;iza RST 8 pretvara u adresu.
750         LD     E,(HL)               ;Zatim uzima 16-bitni broj sa
760         INC    HL                   ;lokacija adresa+1 i adresa+2.
770         LD     D,(HL)               ;Rezultat je u DE registru.
780         RET
790
800 COMTAB: DEFW NAMES-INIT           ;Adresa početka imena komandi.
810 L11:    JP     DASH-INIT           ;Jump instrukcije na početak
820 L12:    JP     MASK-INIT          ;pojedinih potprograma.
830 NAMES:  DEFW "DAS"                ;I tablica sa imenima.Kraj je
840         DEFB "H"+128               ;označen sa 0.
850         DEFW "MAS"
860         DEFB "K"+128,0
870
880 DASH:   CP     2                    ;:DASH RSX komanda ima 2 parametra,
890         JR     NZ,ERLINK            ;inče greška.
900         LD     L,(IX+0)             ;Uzimanje parametra duljine
910         LD     H,(IX+1)             ;linija koje će se crtati.
920         LD     A,H                  ;Duljina mora biti u opsegu
930         OR     L                    ;1..32767
940         JR     Z,RSXERR              ;Inače se prijavljuje
950         BIT    7,H                  ;greška.
960 ERLINK: JR     NZ,RSXERR
970 L13:    LD     (DSHLEN-INIT),HL     ;Pozitivna vrijednost duljine.
980         LD     A,H                  ;Treba izračunati i 0-HL,da
990         CPL                                ;bi se dobila negativna vrijednost
1000        LD     H,A                  ;duljine (jer se linije crtaju
1010        LD     A,L                  ;relativno).
1020        CPL
1030        LD     L,A
1040        INC    HL
1050 L14:   LD     (DSHMLN-INIT),HL     ;Kao i pozitivna i negativna se
1060        LD     L,(IX+2)              ;smješta u memoriju.Dalje treba
1070        LD     H,(IX+3)              ;uzeti adresu STRING DESCRIPTORA,
1080        DEC    HL                    ;koji nam kazuje gdje je niz znakova
1090        LD     A,(HL)                ;koje treba interpretirati.
1100        CP     2                    ;Ispred descriptora se nalazi broj 2,
1110        JR     NZ,RSXERR              ;koji znači da se radi o string
1120        INC    HL                    ;varijabli.Inače greška.Drugi bajt
1130        LD     B,(HL)                ;sadrži duljinu stringa.Treći i
1140        INC    HL                    ;četvrti sadrže apsolutnu adresu
1150        LD     E,(HL)                ;stringa.Majmunski znak

```



1160	INC	HL	;(business a) ispred ovog parametra
1170	LD	D,(HL)	;je obavezan, jer tada se prenosi
1180	EX	DE,HL	;adresa varijable, a ne stvarna
1190	LD	A,B	;vrijednost. Za stringove je to
1200	OR	A	;i jedini način prijenosa. String koji
1210	RET	Z	;nema ni jedan znak ne treba uzimati.
1220	DSHDO:	LD A,(HL)	;Slijedi izvršna petlja DASH rutine.
1230	LD	C,A	;Uzeti znak mora biti u opsegu "0".. "7"
1240	AND	%11111000	;(što se provjeri preko najznačajnijih
1250	CP	"0"	;bitova), inače se prijavi
1260	JR	NZ,RSXERR	;greška.
1270	LD	A,C	;Bitovi 0-2 nam služe za određivanje
1280	AND	3	;smjera, a bit 3 da li se radi o crtanju
1290	PUSH	BC	;ili samo pomaku kurzora.
1300	PUSH	HL	;Spremanje registara za kontrolu petlje
1310	LD	B,A	;na stek. Svaki smjer ima program koji
1320	ADD	A,A	;ga regulira dužine 5 bajtova (LD HL i
1330	ADD	A,A	;JR). Stoga se smjer množi sa 5 i dodaje
1340	ADD	A,B	;početku programa za prvi smjer, a to je
1350	LD	E,A	;program za ulijevo. 5*smjer ne izlazi
1360	LD	D,0	;iz opsega 0..255, pa lakše množimo
1370	L14A:	LD HL,DSHTAB-INIT	;sa A, ali je 16-bitno zbrajanje lakše
1380	ADD	HL,DE	;sa HL i DE registron.
1390	CALL	#1E	;PCHL INSTRUCTION;Ovaj potprogram
1400	BIT	2,C	;simulira CALL (HL). C registar čuva
1410	PUSH	AF	;kod smjera. Bit 2 određuje crtanje.
1420	CALL	NZ,#BBF9	;GRA LINE RELATIVE
1430	POP	AF	;odnosno necrtanje linije.
1440	CALL	Z,#BBC3	;GRA MOVE RELATIVE
1450	POP	HL	;Sada još treba obnoviti registre
1460	POP	BC	;za kontrolu petlje.
1470	INC	HL	;prijeći na slijedeći znak u stringu.
1480	DJNZ	DSHDO	;i obraditi slijedeći znak,
1490	RET		;sve do kraja.
1500			
1510	DSHTAB:	LD HL,(DSHMLN-INIT)	;Ovo su programi za smjerove: lijevo
1520	JR	LDHL0C	;DRAWR (MOVER) -duljina,0
1530	L15:	LD HL,(DSHLEN-INIT)	;desno
1540	JR	LDHL0C	;DRAWR (MOVER) duljina,0
1550	L16:	LD HL,(DSHMLN-INIT)	;dolje
1560	JR	LDDE0C	;DRAWR (MOVER) 0,-duljina
1570	L17:	LD HL,(DSHLEN-INIT)	;gore
1580	LDDE0C:	LD DE,0	;DRAWR (MOVER) 0,duljina
1590	RET		
1600	LDHL0C:	LD DE,0	;DE registar je X, a HL je Y.
1610	EX	DE,HL	
1620	RET		
1630			
1640	AIX0:	LD A,(IX+1)	;Ovaj potprogram uzima parametar
1650	OR	A	;u A. Viši bajt parametra, dakle, mora
1660	LD	A,(IX+0)	;biti 0.
1670	RET	Z	;Ako viši bajt nije 0, RET
1680	POP	AF	;adresa se uništava i javlja se greška.
1690	RSXERR:	LD B,11	;Potprogram za prijavu greške.
1700	L18:	LD HL,EMSG-INIT	;Ovo je adresa poruke od 11 znakova.
1710	RELOOP:	LD A,(HL)	;Slijedi ispis.
1720	CALL	#BB5A	;TXT OUTPUT
1730	INC	HL	
1740	DJNZ	RELOOP	
1750	RET		
1760			
1770	EMSG:	DEFM "RSX error"	
1780	DEFB	10,13	
1790			
1800	MASK:	CP 2	;:MASK RSX komanda ima 2 parametra
1810	JR	Z,MASK2	;ili samo jedan.
1820	DEC	A	
1830	JR	NZ,RSXERR	
1840	MASK1:	CALL AIX0-INIT	;Prihvaćanje parametra maske.
1850	L18A:	LD (mask-INIT),A	;Postavljanje nove maske.
1860	RET		
1870	MASK2:	CALL AIX0-INIT	;Prihvaćanje parametra za prvu
1880	CP	2	;točku u liniji. Mora biti 0 ili 1.
1890	JR	NC,RSXERR	;Inače greška.
1900	L18B:	LD (FRP-INIT),A	;Postavljanje prekidača.
1910	INC	IX	;IX registar se postavlja na slijedeći
1920	INC	IX	;parametar (masku) i nastavak slijedi
1930	JR	MASK1	;kao da je bio samo jedan parametar.
1940			
1950	LINE:	LD BC,SCRWR-INIT	;Mijenjanjem SCR WRITE možemo
1960	LD	(#BDE9),BC	;onemogućiti upis u video memoriju ili
1970	L19:	LD A,(mask-INIT)	;napraviti nešto drugo. Uzima se maska
1980	L20:	LD (maskwk-INIT),A	;i postavlja u radnu varijablu, koju

1990 L21:	LD	A, (FRP-INIT)	;čemo mijenjati. Provjera prekidača za
2000	OR	A	;crtanje prve točke. Ako se crta, posao
2010	JR	Z, JP1c	;je nešto jednostavniji.
2020 L22:	LD	A, (maskwk-INIT)	;Točka koja se ne crta takode mijenja
2030	RRCA		;masku, kao i da se crta.
2040 L23:	LD	(maskwk-INIT), A	;Sada radna varijabla pokazuje na novu.
2050	PUSH	HL	;Spremaju se X i Y kordinate točke do
2060	PUSH	DE	;koje se crta od kurzora na stack.
2070	;Treba saznati boju prve točke. Ima dva slučaja:		
2080 L25:	LD	BC, SCRRD-INIT	;točka jest i točka nije u prozoru.
2090	LD	(#BDE6), BC	;Nastavak ove rutine je drukčiji ako
2100	CALL	#BBC6	;GRA ASK CURSOR; točka jest u prozoru.
2110	CALL	#BDDF	;IND:GRA TEST; Konačno zovemo TEST.
2120 L26:	CALL	BCJPSC-INIT	;Prva točka, dakle, nije u prozoru.
2130	POP	DE	;Obnavlja se SCR READ i X i Y
2140	POP	HL	;koordinata točke do koje se crta.
2150 JP1c:	CALL	JUMPDR-INIT	;Zovemo i samu rutinu koja crta liniju.
2160	JR	LINEex	;Na kraju treba obnoviti SCR WRITE.
2170			
2180 JUMPDR:	LD	BC, (JPDR-INIT)	;Ova rutina vrši skok na program
2190	PUSH	BC	;koji crta liniju u ROMu.
2200	RET		
2210			
2220 BCJPSC:	LD	BC, (JPSCRRD)	;Potprogram za obnavljanje vrijednosti
2230	LD	(#BDE6), BC	;IND:SCR READ
2240	RET		
2250			
2260 SCRRD:	POP	AF	;Prva točka jest u prozoru. RET se
2270	LD	A, C	; "skida" sa stacka.
2280 L27:	CALL	BCJPSC-INIT	;Obnavljanje SCR READ
2290	LD	C, A	;HL=adresa prve točke, C=maska za točku.
2300	AND	(HL)	;Pamćenje točke sa ekrana.
2310	LD	B, A	
2320	POP	DE	;X koordinata
2330	EX	(SP), HL	;Uzimanje Y i spremanje adrese točke.
2340	PUSH	BC	;Sprema se i maska, kao i boja točke.
2350 L28:	CALL	JUMPDR-INIT	;Poziv potprograma u ROMu.
2360	POP	BC	;Vraćanje boje i maske.
2370	POP	HL	;Vraćanje adrese prve točke.
2380	LD	A, C	;Potprogram za liniju je nacrtao
2390	CPL		;prvu točku, ali mi je vraćamo na staro.
2400	AND	(HL)	;Izolacija svih ostalih piksela u bajtu
2410	OR	B	;sa anti-maskom i crtanje točke.
2420	LD	(HL), A	;Dobiveni bajt se vraća u memoriju.
2430 LINEex:	LD	HL, (JPWRM-INIT)	;Obnavljanje IND:SCR WRITE.
2440	LD	(#BDE9), HL	
2450	RET		
2460			
2470 SCRWR:	LD	A, (maskwk-INIT)	;Podmetnuta rutina za pristup memoriji.
2480	CP	#FF	;Ako je maska 255, sve je kao da i
2490	JR	NZ, maskdo	;nema ovog programa.
2500 JPWRIT:	PUSH	HL	;Skok na WRITE rutinu u ROMu.
2510 L29:	LD	HL, (JPWRM-INIT)	
2520	EX	(SP), HL	
2530	RET		
2540			
2550 maskdo:	PUSH	BC	;Spremanje BC
2560	LD	A, C	;A=maska
2570 L30:	LD	(CMEM-INIT), A	
2580 mloopA:	RRCA		;Maska se vrti sve dok se ne nađe
2590	JR	NC, mloopA	;maska za prvi lijevo piksel u bajtu.
2600	PUSH	AF	;Spremanje maske na stack.
2610 L31:	CALL	GETMSK-INIT	;B=broj piksela u bajtu, C=osnovna maska
2620	POP	AF	;Povratak vrijednosti akumulatora.
2630	CP	C	;Da li je ista kao i ona koju
2640	JR	NZ, NOSINGL	;mislimo da smo izdvojili?
2650	POP	BC	;Ako jest, znači da se radi o
2660 L32:	LD	A, (maskwk-INIT)	;okomitom crtanju.
2670	RLCA		;Pomak maske za liniju.
2680 L33:	LD	(maskwk-INIT), A	
2690	JR	C, JPWRITE	;Ako treba crtati točku.
2700	PUSH	BC	;Slučaj kada točku treba nacrtati
2710 PAP1:	LD	A, (#B339)	;sa bojom gpapera.
2720	LD	B, A	
2730 L34:	CALL	JPWRITE-INIT	
2740	POP	BC	
2750	RET		
2760			
2770 NOSING:	PUSH	DE	;Linija se crta vodoravno.
2780 L35:	LD	A, (CMEM-INIT)	;E=maska za izolaciju dijela bajta
2790	LD	E, A	; (ili čak cijelog)
2800 L36:	LD	A, (maskwk-INIT)	;D=radna maska za liniju.



```

2810      LD      D,A
2820 NSLP:    LD      A,C
2830      AND     E
2840      CP      C
2850      JR      NZ,NNEXT
2860 PEN:    LD      A,(#B338)
2870      RLC     D
2880      JR      C,NSLPC1
2890 PAP2:   LD      A,(#B339)
2900 NSLPC1: LD      B,A
2910 L37:    CALL   JPWRITE-INIT
2920 NNEXT:  RRC     C
2930      JR      NC,NSLP
2940      LD      A,D
2950 L38:    LD      (maskwk-INIT),A
2960      POP     DE
2970      POP     BC
2980      RET
2990
3000 GETMSK: CALL   #BC11
3010      LD      BC,#01AA
3020      RET     C
3030      LD      BC,#0388
3040      RET     Z
3050      LD      BC,#0780
3060      RET
3070
3080 mask:   DEFB 255
3090 maskwk: DEFB 0
3100 DSHLEN: DEFW 0
3110 DSHMLN: DEFW 0
3120 KERSP:  DEFS 4
3130 FRP:    DEFB 0
3140 JPWRM:  DEFW 0
3150 JPSCRR: DEFW 0
3160 JPDR:   DEFW 0
3170 CMEM:   DEFB 0
3180
3190 ;

```

;Sada se ispituje da li se radi o  
;pikselu kojeg treba crtati ili ne.  
;Ako ovaj piksel ne treba crtati,  
;prelazi se na sljedeći.  
;Uzimanje gpena.  
;Pomak maske.  
;Ako ne treba uzeti gpaper (jer se crta  
;gpen).  
;B=boja kojom se crta,C=maska za piksel  
;Rutina za pristup memoriji u ROMu.  
;Rotiranje maske za bajt koji se  
;obrađuje.Kada dođe do prijenosa,posao  
;je gotov.Treba spremi radnu masku.  
;Smještanje nove radne maske.  
;Povratak vrijednosti DE,koji se ne  
;prlja u originalnoj rutini.

;SCR GET MODE;A=mode:CP 1  
;MODE 0:0..1 pikseli,X10101010 mask  
;MODE 1:0..3 pikseli,X10001000 mask  
;MODE 2:0..7 pikseli,X10000000 mask

;Maska za linije  
;Radna maska  
;Duljina linije  
;-Duljina linije  
;Prostor za vezanje RSX komandi  
;Prekidač prve točke u liniji  
;Adresa IND:SCR WRITE  
;Adresa IND:SCR READ  
;Adresa IND:GRA LINE  
;Radne varijable

by Siniša Jagodić, 1988.

#### Listing 6

```

10 N=HIMEM-632:MEMORY N:LOAD"!DASHMASK.BIN",N+1:CLOSEIN:CALL N+1:MEMORY N+214:NE
W

```

#### Listing 7

```

10 MEMORY &7FFF:FOR N=0 TO 19:S=0:READ A$:FOR M=1 TO 64 STEP 2:Z=VAL("&"MID$(A$,
,M,2)):S=S+Z:POKE &8000+N*32+(M-1)/2,Z:NEXT:READ Q:I
F S<>Q THEN PRINT"Greska u liniji broj"N*10+100:END ELSE NEXT
20 SAVE"DASHMASK.BIN",B,&8000,&278
100 DATA 21E1E9223000F7EB212100194E234679B0287DE5606919E54E2346606919444D,3029
110 DATA E1712370E12318E48D00900099009F00A800AC00B200B600BA00C000C300CF00,3074
120 DATA D200D500F300FD002A01410146014B01500184018B018701870192019B01A201,2235
130 DATA A501A801AE01B201B701C401C901CF01D501DF01E801F201F901010208020F02,2674
140 DATA 17021B0225022B022F0243024B0200002AE3BD226E02219A0122E3BD2AE9BD22,2073
150 DATA 6A022AE6BD226C02CD06B92AE2BBCDC800ED5338022AE8BBCDC800ED532102ED,3805
160 DATA 533F02CD09B921650201CF00C3D1BCCBBC235E2356C9D700C3E000C37C014441,3412
170 DATA 53C84D4153CB00FE02200CDD6E00DD66017CB52876CB7C20722261027C2F677D,3134
180 DATA 2F6F23226302DD6E02DD66032B7EFE0220592346235E2356EB78B7C87E4FE6F8,3309
190 DATA FE30204779E803C5E5478787805F160021400119CD1E00CB51F5C4F9BBF1CCC3,3930
200 DATA BBE1C12310D6C92A630218112A6102180C2A630218032A6102110000C9110000,1978
210 DATA EBC9DD7E01B7DD7E00C8F1060B2171017ECD5ABB2310F9C9525358206572726F,3753
220 DATA 720A0DFE02280A3D20E1CD5B01325F02C9CD5B01FE0230D3326902DD23DD2318,2911
230 DATA E901F801ED43E9BD3A5F023260023A6902B7281B3A60020F326002E5D501DC01,2910
240 DATA ED43E6BDCDC6BBCDDFBDCCDD301D1E1CDDC011824ED4B6E02C5C9ED4B6C02ED43,4800
250 DATA E6BDC9F179CDD3014FA647D1E3C5CDCD01C1E1792FA6B0772A6A0222E9BDC93A,4671
260 DATA 6002FEFF2006E52A6A02E3C9C5793270020F30FDF5CD5002F1B92014C13A6002,3609
270 DATA 0732600238E0C53A39B347CDFF01C1C9D53A70025F3A60025779A3B9200E3A38,3204
280 DATA B3CB0238033A39B347CDFF01CB0930E97A326002D1C1C9CD11BC01AA01D80188,3570
290 DATA 03C8018007C9FF000000000000000000000000000000000000000000000000,0795

```

#### Listing 8

```

10 MODE 1:A$="465775641564724657":;MASK,&X10010011:PLOT -10,-10,1
20 FOR N=30 TO 300:MOVE N,N:;DASH,@A$,N:NEXT:CALL &BB06
30 MODE 1:;MASK,&X11001100:FOR N=10 TO 400 STEP 20:MOVE 320,200:;DASH,@A$,N:NEXT
:CALL &BB06
40 DEG:MODE 1:;MASK,255:FOR N=10 TO 450 STEP 20:PLOT N,200+SIN(N)*50,RND*3+1:;DA
SH,@A$,N/5:NEXT:CALL &BB06
50 MODE 1:A$="4657":FOR N=0 TO 359:PLOT 320+COS(N)*280,200+SIN(N)*150,RND*3+1:;D
ASH,@A$,RND*30+5:NEXT:CALL &BB06

```