

```

10 ' Programa 1
20 ' RSX para impressao em multicolor
40 '
50 h=HIMEM-1:IF PEEK(0)=0 THEN 80
60 POKE 0,0:h=h-192:MEMORY(h-1)
70 lin=1000:enderesso=h:FOR t=1 TO 192 STEP 8
80 soma=0:FOR b=0 TO 7
90 READ a$:a$=UPPER$(a$):n=1
100 IF LEN(a$)<>2,THEN 370
110 b$=MID$(a$,n,1)
120 GOSUB 350:IF a=0 THEN 370
130 n=n+1:IF n=2 THEN 120
140 p=VAL("&" + a$):POKE enderesso,p
150 enderesso=enderesso-1:soma=soma+p
160 NEXT b:READ a
170 IF a<>soma THEN 370
180 PRINT "LINHA";lin;"OK":lin=lin+10
190 NEXT t
200 r=h+9:a=INT(r/256):b=r-(a*256)
210 POKE h+1,b:POKE h+2,a
220 r=h+17:a=INT(r/256):b=r-(a*256)
230 POKE h+4,b:POKE h+5,a
240 r=h+21:a=INT(r/256):b=r-(a*256)
250 POKE h+9,b:POKE h+10,a
260 r=h+30:a=INT(r/256):b=r-(a*256)
270 POKE h+41,b:POKE h+42,a
280 POKE h+63,b:POKE h+64,a
290 r=h+31:a=INT(r/256):b=r-(a*256)
300 POKE h+122,b:POKE h+123,a
310 r=h+194:a=INT(r/256):b=r-(a*256)
320 POKE h+31,b:POKE h+32,a
330 CALL h
340 END
350 a=INSTR("0123456789ABCDEF",b$)
360 RETURN
370 PRINT "ERRO NA LINHA";lin;"!":END
1000 DATA 01,29,4e,21,31,4e,c3,d1, 684
1010 DATA bc,35,4e,18,18,00,18,11, 408
1020 DATA 00,00,00,00,00,4d,50,55, 242
1030 DATA d4,47,50,55,d4,00,00,30, 708
1040 DATA 75,f5,af,18,03,f5,3e,01, 872
1050 DATA 32,3e,4e,f1,a7,c8,47,f5,1114
1060 DATA dd,23,dd,23,10,fa,cd,11,1000
1070 DATA bc,47,3e,02,90,f5,3a,3e, 832
1080 DATA 4e,a7,28,0a,cd,78,bb,2d, 852
1090 DATA 25,cd,1a,bc,18,1b,cd,c6, 910
1100 DATA bb,b7,cb,1c,cb,1d,cd,11,1055
1110 DATA bc,f1,f5,a7,28,08,47,b7,1143
1120 DATA cb,1a,cb,1b,10,f9,cd,1d, 958
1130 DATA bc,f1,c1,c5,f5,dd,2b,dd,1549
1140 DATA 2b,4f,06,03,80,47,79,e5, 680
1150 DATA ed,5b,3f,4e,dd,6e,00,26, 838
1160 DATA 00,29,10,fd,19,eb,e1,87, 930
1170 DATA 20,01,3c,47,c5,e5,06,08, 604
1180 DATA 3a,05,00,a7,28,09,fe,01, 534
1190 DATA 28,0b,fe,02,1a,18,08,1a, 391
1200 DATA a7,28,05,18,02,1a,ae,77, 557
1210 DATA cd,26,bc,13,10,e2,e1,cd,1122
1220 DATA 20,bc,c1,10,d7,f1,c1,10,1094
1230 DATA b2,c9,00,00,00,00,00,00, 379

```

```

10 'Programa 2 ----- DESENHADOR DE CARACTERES
20 INPUT "Modo";mm:md=1
30 IF mm<>1 THEN md=2
40 SPEED KEY 10,1:MODE mm
50 WINDOW#1,1,40/md,24,25
60 MEMORY 19999
70 q=1:BORDER 0
80 PEN 1:tt=TEST(x,398-y):LOCATE INT(x/(2*md))+10,INT(y/2)+1:PRINT CHR$(203)
90 k$=INKEY$:IF k$="" THEN 80
100 kk=ASC(k$)
110 LOCATE INT(x/(2*md))+10,INT(y/2)+1:PEN tt:PRINT CHR$(143):PEN q
120 x=x-(2*md)*((kk=&F3 AND x<(14*md))-(kk=&F2 AND x<>0))
130 y=y-2*((kk=&F1 AND y<14)-(kk=&F0 AND y<>0))
140 IF k$="p" OR k$="P" THEN PEN 1:INPUT#1,"Tinta";q:PEN q:CLS#1
150 IF kk=224 THEN te=1-(te=1):PEN 1:LOCATE 1,16:PRINT"OFF":IF te=1 THEN LOCATE
1,16:PRINT"ON "
160 LOCATE 1,15:PEN 1:PRINT "Tinta ";q;" ":PEN q
170 IF k$="r" OR k$="R" THEN 240
180 IF k$="s" OR k$="S" THEN 220
190 IF k$="q" OR k$="Q" THEN 280
200 IF te=1 THEN LOCATE INT(x/(2*md))+10,INT(y/2)+1:PRINT CHR$(143):PLOT x,398-y
,q
210 GOTO 80
220 PEN 1:INPUT#1,"Sprite";a:a=a*16*md+20000:PEN q:CLS#1
230 c=49152:FOR b=0 TO (2*md)-1:FOR n=0 TO 7:POKE a+8*b+n,PEEK(c+(n*2048)+b):NEX
T n:NEXT b:GOTO 80
240 PEN 1:INPUT#1,"Sprite";a:a=a*16*md+20000:PEN q:CLS#1
250 c=49152:FOR b=0 TO (2*md)-1:FOR n=0 TO 7:POKE c+(n*2048)+b,PEEK(a+8*b+n):NEX
T n:NEXT b:GOSUB 260:GOTO 80
260 FOR b=0 TO 14 STEP 2:FOR n=0 TO 14*md STEP(md*2):tt=TEST (n,398-b):LOCATE IN
T(n/(2*md))+10,INT(b/2)+1:PEN tt:PRINT CHR$(143):NEX
T n:NEXT b:PEN 1
270 RETURN
280 PEN 1:INPUT#1,"Gravar caracteres";r$:CLS#1:r$=UPPER$(r$):IF LEFT$(r$,1)<>"S"
THEN 200
290 INPUT#1,"Quantos";s:CLS#1
300 INPUT#1,"Com que nome";nome$:CLS#1:s=s*16*md:SAVE nome$,b,20000,s:GOTO 200

```



```

10 ;          M U L T I C O R
20 ;
30 ; Introduzir os comandos residentes INPUT e IGPOT
40 ;
50          LD    BC,COMAND          ;addresso da tabela de comandos
60          LD    HL,BYTES          ;addresso de 4 bytes para o S.O
70          JP    #BCD1             ;
80 COMAND:   DEFW #ASCf              ;addresso da tabela de nomes
90          JR    ENTRY2            ;salto para a rotina "MPUT"
100         NOP
110         JR    ENTRY1            ;salto para a rotina "GPUT"
120         DEFB #0
130 BYTES:   DEFB #0,#0,#0,#0        ;area para o uso do S.O.
140         DEFB "M","P","U",#D4     ;'MPU','T'+#80'
150         DEFB "G","P","U",#D4     ;'GPU','T'+#80'
160         DEFB #0                  ;marcador de fim da tabela
170 FLAG:    DEFB #0                ;indica se se esta a processar MPUT ou
180 ;                                ;GPUT
190 DIRDAT:  DEFW #A67C              ;addresso dados caracteres
200 ;
210 ;aqui comeca a rotina GPUT
220 ;
230 ENTRY1:  PUSH AF                ;preserva no. de caracteres
240          XOR    A                ;apaga acumulador
250          JR    COMUM
260 ;
270 ;aqui comeca a rotina MPUT
280 ;
290 ENTRY2:  PUSH AF                ;preserva no. de caracteres
300          LD     A,#1              ;se e MPUT, flag=1 - se e GPUT
310 COMUM:   LD     (FLAG),A          ;flag=0
320          POP    AF                ;recupera no. de caracteres
330          AND    A                ;se nao ha caracteres retorna ao BASIC
340          RET    Z
350          LD     B,A                ;se ha inicializa contador
360          PUSH  AF                ;grava no. de caracteres
370 CICLO1:  INC     IX                ;e actualiza IX para que aponte
380          INC     IX                ;para tras do primeiro caracter
390          DJNZ   CICLO1
400          CALL  #BC11              ;averigua o modo do ecran
410          LD     B,A                ;fica 2-modo
420          LD     A,#2
430          SUB    B
440          PUSH  AF                ;preserva o resultado
450          LD     A,(FLAG)           ;carrega a flag (1: MPUT - 0: GPUT)
460          AND    A
470          JR    Z,GPUT2            ;se estamos em GPUT, salta
480          CALL  #BB7B              ;averigua posicao do CURSOR de texto
490          DEC    L                  ;converte as coordenadas logicas
500          DEC    H                  ;a fisicas e averigua o addresso
510          CALL  #BC1A              ;correspondente da memoria
520          JR    MPUT2              ;do ecran
530 GPUT2:   CALL  #DBC6              ;averigua coordenados do CURSOR grafico

540          OR     A
550          RR     H                  ;apaga o acarreo
560          RR     L                  ;converte coordenada y de "usuario"
570          CALL  #BC11              ;a coordenada y de "base"
580          POP    AF                ;averigua o modo do ecran
590          PUSH  AF                ;recupera 2-modo
600          AND    A                ;e torna a preserv-lo
610          JR    Z,PIXEL            ;testa se e 0 (modo 2)
620          LD     B,A                ;se e modo 2, salta
630 CICLO2:  OR     A                ;se nao, passa 2-modo a B
640          RR     D                  ;apaga o acarreo
650          RR     E                  ;converte coordenada X de "usuario" a
660          DJNZ   CICLO2            ;coordenada X de "base"
670 PIXEL:   CALL  #BC1D              ;calcula addresso de pixel no ecran
680 MPUT2:   POP    AF                ;recupera 2-modo
690          POP    BC                ;recupera no. de caracteres
700 OUTRO:   PUSH  BC                ;grava no. de caracteres
710          PUSH  AF                ;grava 2-modo

```

720	DEC IX	;aponta o caracter
730	DEC IX	
740	LD C,A	;carrega em C 2-modo
750	LD B,#3	
760	ADD A,B	;soma-lhe 3
770	LD B,A	;passa 2-modo+3 a B
780	LD A,C	;recupera 2-modo
790	PUSH HL	;preserva aderesso do pixel
800	LD DE,(DIRDAT)	;DE=#a67c principio dos dados
810	LD L,(IX+0)	;de caracter
820	LD H,#0	
830	CICLO3: ADD HL,HL	;calcula aderesso em que se encontra
840	DJNZ CICLO3	
850	ADD HL,DE	
860	EX DE,HL	;passa aderesso a DE
870	POP HL	;recupera aderesso do pixel
880	ADD A,A	;duplica 2-modo para calcular a largura
890	JR NZ,LARG	;do caracter.Se e modo 2,largura=1
900	LARG: INC A	;largura de caracter a B
910	COLUMN: PUSH BC	;preserva largura e 2-modo
920	PUSH HL	;preserva aderesso do pixel
930	LD B,#8	;inicializa contador
940	CICLO4: LD A,(#5)	;carrega o modode impressao
950	AND A	;prova se e transparente
960	JR Z,TRANSP	
970	CP #1	;prova se e um (XOR)
980	JR Z,MODXOR	;se nao,e modo normal
990	CP #2	
1000	LD A,(DE)	
1010	JR NORMAL	
1020	TRANSP: LD A,(DE)	;carrega primeiro byte do caracter
1030	AND A	;se e zero, passa a NORM2
1040	JR Z,NORM2	
1050	JR NORMAL	;e se nao a NORMAL
1060	MODXOR: LD A,(DE)	;carrega byte do caracter
1070	XOR (HL)	;XOR com o do ecran
1080	NORMAL: LD (HL),A	;envia byte para o ecran
1090	NORM2: CALL #BC26	;calcula aderesso do byte interior do
1100	;	;ecran
1110	INC DE	;aponta para o seguinte byte do caracter
1120	DJNZ CICLO4	;repete para os oito bytes
1130	POP HL	;recupera aderesso da esquina superior
1140	;	;do caracter.
1150	CALL #BC20	;calcula aderesso da coluna seguinte
1160	POP BC	;recupera altura do caracter
1170	DJNZ COLUMN	;repete ate completar o caracter
1180	POP AF	;recupera 2-modo
1190	POP BC	;recupera no. de caracteres
1200	DJNZ OUTRO	;repete para todos os caracteres
1210	RET	;volta ao BASIC