

SOMMAIRE

Daiclic 17

- Caractères en RAM : 7 p = p 1
par Di Marco Emmanuel
- Remember with REMTest : 8 p = p 8
par Di Marco Emmanuel
- Remarques en SPL : 2 p = p 16
par Claudius Morin
- Traduction d'instructions du
X-Basic Allemand : 10 p = p 18
par Menier Joël

Chers Amis :

Encore un appel au secours !

Depuis Août me voici sous les drapeaux

⇒ diminution de mes revenus et surtout de mon temps libre

↳ pas d'argent pour dépanner mon imprimante.

↳ fort peu de temps à consacrer au club et à la revue.

Je vous annonce que je ne reprendrais pas seul le flambeau du club l'an prochain.

Que deviendra l'unique Bibliothèque consacrée au DAI.

Plusieurs kilos de documents... en tant d'années.

Je les donne à celui qui voudrait poursuivre une quelconque activité conviviale (contre remboursement des frais de port.)

Si je n'ai pas plus d'articles d'ici Décembre, j'ai bien peur que mon dernier numéro de Daiclic ne se résume à une simple lettre d'adieux.

Ecrasons une dernière larme sur toutes ces années investies dans une meilleure connaissance de soi, au travers de cette fabuleuse machine, hélas mais heureusement dépassée.

Expérience oh combien gratifiante.

Bruno Delannay

```
10 CLEAR #3700
20 MODE 0
30 COLORT 8 0 0 0
40 COLORG 8 0 0 0
50 PRINT CHR*(12);
60 MODE 1A
65 POKE #2A4,#4B
70 DRAW 0,0 9,0 0
80 DRAW 0,0 0,11 0
90 DRAW 9,0 9,11 0
100 DRAW 0,11 9,11 0
110 X%=1.0:Y%=1.0:ADDEB%=#40B0:CHAR%=0.0
115 GOSUB 2000:REM EXTRACTION CHAR
120 MEMC%=SCRN(X%,Y%)
130 DOT X%,Y% 0
140 DOT X%,Y% 8
150 DOT X%,Y% MEMC%
160 G%=GETC:IF G%=0.0 GOTO 130
170 IF G%=16.0 GOTO 300
180 IF G%=17.0 GOTO 400
190 IF G%=18.0 GOTO 500
200 IF G%=19.0 GOTO 600
205 IF G%=9.0 GOTO 1000:REM SAVE CHAR
206 IF G%=13.0 GOTO 3000:REM AUTRE CHAR
210 IF G%<>32.0 GOTO 130
220 IF MEMC%=8.0 THEN DOT X%,Y% 0:GOTO 120
230 DOT X%,Y% 8
240 GOTO 120
300 Y%=Y%+1.0
310 IF Y%>10.0 THEN Y%=10.0
320 GOTO 120
400 Y%=Y%-1.0
410 IF Y%<1.0 THEN Y%=1.0
420 GOTO 120
500 X%=X%-1.0
510 IF X%<1.0 THEN X%=1.0
520 GOTO 120
600 X%=X%+1.0
610 IF X%>8.0 THEN X%=8.0
620 GOTO 120
1000 ADRES%=ADDEB%+10.0*CHAR%
FOR V%=1.0 TO 10.0
1005 1 VA%=0.0
1010 1 FOR U%=1.0 TO 8.0
1020 2 Z%=SCRN(U%,V%)
1030 2 IF Z%=8 GOTO 1100
1035 2 GOSUB 5000
1040 2 VA%=VA%+H%
1100 1 NEXT U%
1110 1 POKE ADRES%+V%-1,VA%
1120 NEXT V%
1130 GOTO 3000
2000 ADRES%=ADDEB%+10.0*CHAR%
2010 FOR V%=1.0 TO 10.0
2020 1 Z%=PEEK(ADRES%+V%-1)
```

PAGE 2 ---

-- FLST

```

2030 1   FOR UX=1.0 TO 8.0
2035 2       GOSUB 5000
2040 2       W%=Z% IAND H%
2045 2       IF W%=0.0 GOTO 2060
2050 2       DOT UX,V% 0:GOTO 2065
2060 2       DOT UX,V% 8
2065 1   NEXT UX
2070     NEXT V%
2080     PRINT CHR$(12);"CHAR = ";HEX$(CHAR%);"H"
2090     RETURN
3000     PRINT CHR$(12);
3002     INPUT "VALEUR ASCII ";VASC%
3005     IF VASC%<0 OR VASC%>127 GOTO 3000
3010     CHAR%=VASC%
3020     PRINT CHR$(12);
3030     GOTO 115
5000     ON UX GOTO 5010,5020,5030,5040,5050,5060,5070,5080
5010     H%=128:RETURN
5020     H%=64:RETURN
5030     H%=32:RETURN
5040     H%=16:RETURN
5050     H%=8:RETURN
5060     H%=4:RETURN
5070     H%=2:RETURN
5080     H%=1:RETURN
10000    FOR ZX=#4000 TO #5000
10010        POKE ZX,0
10020    NEXT ZX
10030    END

```

SPL V1.1 PAGE 1

```

0000     TITL      'CHAR GRAPHIC MANAGEMENT ROUTINE'
0000     ORG      2000H
2000     F3      START
2001     F5      DI
2002     E5      PUSH PSW
2003     D5      PUSH H
2004     C5      PUSH D
2005     3A9D00  LDA      9DH
2008     FFFF    CPI      0FFH
200A     CAC020  JZ      EXIT
200D     E601    ANI      1H
200F     CA2720  JZ      S1
2012     2A9200  LHL    92H
2015     EB      XCHG
2016     2A9900  LHL    99H
2019     CD14DE  CALL    0DE14H
201C     C2C020  JNZ     EXIT
201F     3A9D00  LDA      9DH
2022     E6FE    ANI      0FEH
2024     EF      RST    5
2025     18      DB      18H
2026     F3      DI
2027     3A9600  S1     LDA      96H
202A     D60A    SUI      0AH
202C     6F      MOV    L,A
202D     2600    MVI    H,0H
202F     3A02F9  LDA      0F902H
2032     2F      CMA
2033     5F      MOV    E,A
2034     16FF    MVI    D,0FFH
2036     19      DAD    D
2037     23      INX    H
2038     7C      MOV    A,H
2039     17      RAL
203A     DAC020  JC      EXIT
203D     3A9800  LDA      98H
2040     D602    SUI      2H
2042     37      STC
2043     3F      CMC
2044     1F      RAR
2045     3D      DCR    A
2046     2101F9  LXI    H,0F901H
2049     BE      CMP    M
204A     FAC020  JM      EXIT
204D     3A01F9  LDA      0F901H
2050     2F      CMA
2051     5F      MOV    E,A
2052     16FF    MVI    D,0FFH
2054     2A8800  LHL    88H
2057     19      DAD    D
2058     19      DAD    D
2059     3A9800  LDA      98H
205C     5F      MOV    E,A
205D     1600    MVI    D,0H
205F     3A02F9  LDA      0F902H
2062     3C      INR    A
2063     19      S2     DAD    D
2064     3D      DCR    A

```

;SAUVEGARDE LES REGISTRES

;TEST SI MODE<>0

;EXIT SI MODE 0

;TEST SI SPLIT

;PAS SPLIT => GOTO S1

;TEST SI MODE SPLIT FIXE

;EXIT SI SPLIT FIXE

;SINON MET EN

; MODE NORMAL

;A=NOMBRE DE LIGNES-10

; DANS CE MODE

;LIT LA LIGNE QUE L'ON

; DESIRE ATTEINDRE

; ET PREND LE COMPLEMENT

; LE STOCKE DANS DE

; ET LE SOUSTRAIT A HL

;SI LE BIT7 DE HL EST

; EGAL A 1 ALORS EXIT

; CAR LIGNE>NBLMODE

;A=NBBYTE DANS LIGNE

;CALCULE LE NOMBRE DE BYTES

; PIXELS DANS LIGNE

;TESTE SI POSX

; DESIREE <=POSMAX

;EXIT SI TROP GRAND

;LIT POSX

;DE = POSX INVERSEE

;LIT DEBUT ZONE GRAPHIQUE

;HL=DEBUT-POSX-POSX-2

;A=NBBYTE DANS LIGNE

;DE=NBBYTE DANS LIGNE

;A=LIGNE DESIREE+1

;ADDRHL=ADDRHL+A

SPL V1.1 PAGE 2

CHAR GRAPHIC MANAGEMENT ROUTINE

```

2065 C26320      JNZ      S2          ;SI LIGNE<>0 CONTINUE
2068 2204F9      SHLD     0F904H     ;SAUVE ADDR EN F904/5
206E 13          INX D
206C D5          PUSH D          ;SAUVE NBYTE+1 SUR STACK
206D 3A03F9      LDA      0F903H
2070 E67F        ANI      7FH        ;A=ASC(CHAR) IAND #7F
2072 5F          MOV E,A
2073 1600        MVI D     0H        ;DE=A
2075 62          MOV H,D
2076 6B          MOV L,E          ;HL=DE=A
2077 29          DAD H
2078 29          DAD H
2079 29          DAD H
207A 19          DAD D
207B 19          DAD D          ;HL=10*A
207C 11D020      LXI D     TABL
207E 19          DAD D          ;HL=<F00> + 10*A
2080 2206F9      SHLD     0F906H     ;SAUVE EN F906/7
2083 D1          POP D
2084 060A        MVI B     0AH
2086 3A9D00      LDA      9DH        ;TEST SI MODE 4/16 COULEURS
2089 E602        ANI      2H        ;SI MODE 4 COULEURS GOTO S5
208B C2A920      JNZ      S5        ;SINON 16 COULEURS
208E 2A06F9 S3   LHLD     0F906H     ;TRANSFERT BYTE DE LA
2091 7E          MOV A,M          ; MEMOIRE EN A
2092 23          INX H          ;CALCULE NOUVELLE ADRESSE
2093 2206F9      SHLD     0F906H     ; ET SAUVEGARDE
2096 2F          CMA
2097 2A04F9      LHLD     0F904H     ;ECRIT SUR ECRAN
209A 77          MOV M,A
209B 2B          DCX H
209C 3680        MVI M     80H
209E 19          DAD D          ;CALCULE NOUVELLE ADRESSE
209F 2204F9      SHLD     0F904H     ; ET SAUVEGARDE
20A2 05          DCR B
20A3 C28E20      JNZ      S3        ;CONTINUE SI<>0
20A6 C3C020      JMP      EXIT      ; SINON EXIT
20A9 2A06F9 S5   LHLD     0F906H     ;IDEM CI-DESSUS
20AC 7E          MOV A,M          ; MAIS 4 COULEURS
20AD 23          INX H
20AE 2206F9      SHLD     0F906H
20B1 2A04F9      LHLD     0F904H
20B4 77          MOV M,A
20B5 2B          DCX H
20B6 3600        MVI M     0H
20B8 19          DAD D
20B9 2204F9      SHLD     0F904H
20BC 05          DCR B
20BD C2A920      JNZ      S5
20C0 C1          EXIT      ;REPRISE REGISTRES
20C1 D1          POP D
20C2 E1          POP H
20C3 F1          POP PSW
20C4 FB          EI
20C5 C9          RET          ;ET RETOUR BASIC/MLP
20C6 FFFF        DW      0FFFFH
20C8 FFFF        DW      0FFFFH
20CA FFFF        DW      0FFFFH

```

SPL V1.1 PAGE 3

CHAR GRAPHIC MANAGEMENT ROUTINE

```

20CC FFFF        DW      0FFFFH
20CE FFFF        DW      0FFFFH
20D0 00          TABL     DB      0H
20D1          END      END

```

```

END :20D1 EXIT :20C0 S1 :2027 S2 :2063 S3 :208E S5
START :2000 TABL :20D0

```

```

2000 F3 F5 E5 D5 C5 3A 9D 00 FE FF DA C0 20 E6 01 CA
2010 27 20 2A 92 00 EB 2A 99 00 CD 14 DE C2 C0 20 3A
2020 9D 00 E6 FE EF 18 F3 3A 96 00 D6 0A 6F 26 00 3A
2030 02 F9 2F 5F 16 FF 19 23 7C 17 DA C0 20 3A 98 00
2040 D6 02 37 3F 1F 3D 21 01 F9 BE FA C0 20 3A 01 F9
2050 2F 5F 16 FF 2A 88 00 19 19 3A 98 00 5F 16 00 3A
2060 02 F9 3C 19 3D C2 63 20 22 04 F9 13 D5 3A 03 F9
2070 E6 7F 5F 16 00 62 6B 29 29 29 19 19 11 D0 20 19
2080 22 06 F9 D1 06 0A 3A 9D 00 E6 02 C2 A9 20 2A 06
2090 F9 7E 23 22 06 F9 2F 2A 04 F9 77 2B 36 80 19 22
20A0 04 F9 05 C2 8E 20 C3 C0 20 2A 06 F9 7E 23 22 06
20B0 F9 2A 04 F9 77 2B 36 00 19 22 04 F9 05 C2 A9 20
20C0 C1 D1 E1 F1 FB C9 FF FF FF FF FF FF FF FF FF FF
>

```

```

10 REM TEST DE LA ROUTINE CHAR GRAPHIC
20 CLEAR 256
30 COLORT 8 0 0 0
40 MODE 0
50 COLORG 8 0 0 0
60 MODE 5
65 PRINT "ESSAI":WAIT TIME 50
70 FOR Y=0.0 TO YMAX STEP 10.0
80 FOR X=0.0 TO XMAX/7.0
90 POKE #F901,X
100 POKE #F902,Y
110 POKE #F903,#67
120 CALLM #2000
130 NEXT X
140 NEXT Y
150 G=GETC:IF G=0.0 GOTO 150
160 MODE 0
170 END

```



```

0000          TITL      'RENUMBER WITH REMTEST (DAICLIC NO 15)
0000          :-----
0000          :  MODE D'EMPLOI :
0000          :  *AZ=(PREMIERND SHL 16)+INCREMENT
0000          :  *CALLM #A000,AZ
0000          :  LES INSTRUCTIONS 'REM' SEULES SUR
0000          :  UNE LIGNE PEUVENT ETRE DOTEES D'UN
0000          :  NUMERO DE LIGNE POUR LA RENUMEROTATION
0000          :  CE NUMERO EST ATTRIBUE A LA LIGNE
0000          :  S'IL EST PLUS GRAND QUE CELUI EXISTANT
0000          :  L'INSTRUCTION SE PRESENTE AINSI :
0000          :  123 REM 1000
0000          :  124 REM 10 ESSAI
0000          :  125 REM 700000 TEST
0000          :  LA TROISIEME LIGNE EST IGNOREE CAR
0000          :  LE NUMERO DE LIGNE EST TROP GRAND
0000          :-----
0000          ORG      5000H
5000  F3  RENUM  DI
5001  F5          PUSH PSW
5002  C5          PUSH B
5003  D5          PUSH D
5004  E5          PUSH H
5005  3EFF       MVI A    0FFH
5007  EF          RST 5
5008  18          DB      18H
5009  3E0C       MVI A    0CH
500B  EF          RST 5
500C  03          DB      3H
500D  3E01       MVI A    1H
500F  323101     STA      131H
5012  211253     LXI H   COMM1
5015  CDD4DA     CALL     0DAD4H
5018  CDDAD6     CALL     0D6DAH
501B  F3          DI
501C  FEFF       CPI      0FFH
501E  CA0451     JZ       EXIT
5021  3A4000     LDA      40H
5024  320D54     STA      RESERV
5027  3E30       MVI A    30H
5029  CD08D8     CALL     0D808H
502C  E1          POP H
502D  E5          PUSH H
502E  56          MOV D,M
502F  23          INX H
5030  5E          MOV E,M
5031  23          INX H
5032  EB          XCHG
5033  7C          MOV A,H
5034  B5          ORA L
5035  C24150     JNZ      SUITE1
5038  216E53     LXI H   COMM2
503B  CDD4DA     CALL     0DAD4H
503E  C30451     JMP      EXIT
5041  221054 SUITE1 SHLD     FIRST
5044  EB          XCHG
5045  56          MOV D,M
5046  23          INX H

```

```

5047  5E          MOV E,M
5048  EB          XCHG
5049  7C          MOV A,H
504A  B5          ORA L
504B  C25750     JNZ      SUITE2
504E  219353     LXI H   COMM3
5051  CDD4DA     CALL     0DAD4H
5054  C30451     JMP      EXIT
5057  220E54 SUITE2 SHLD     IN
505A  2A1054 TESTL LHL D   FIRST
505D  44          MOV B,H
505E  4D          MOV C,L
505F  2A9F02     LHL D   29FH
5062  EB          XCHG
5063  2AA102     LHL D   2A1H
5066  2B          DCX H
5067  EB          XCHG
5068  D5          TEST10 PUSH D
5069  7E          MOV A,M
506A  23          INX H
506B  E5          PUSH H
506C  CD0A51     CALL     TSTREM
506F  60          MOV H,B
5070  69          MOV L,C
5071  EB          XCHG
5072  2A0E54     LHL D   IN
5075  19          DAD D
5076  D28450     JNC      TEST20
5079  21BC53     LXI H   COMM4
507C  CDD4DA     CALL     0DAD4H
507F  E1          POP H
5080  D1          POP D
5081  C30451     JMP      EXIT
5084  44          TEST20 MOV B,H
5085  4D          MOV C,L
5086  E1          POP H
5087  D1          POP D
5088  CD30DE     CALL     0DE30H
508B  CD14DE     CALL     0DE14H
508E  DA6850     JC       TEST10
5091  2AA302 REN1  LHL D   2A3H
5094  221254     SHLD     PTABLE
5097  2A1054     LHL D   FIRST
509A  44          MOV B,H
509B  4D          MOV C,L
509C  2A9F02     LHL D   29FH
509F  EB          XCHG
50A0  2AA102     LHL D   2A1H
50A3  2B          DCX H
50A4  EB          XCHG
50A5  D5          REN010 PUSH D
50A6  EB          XCHG
50A7  2A1254     LHL D   PTABLE
50AA  EB          XCHG
50AB  7E          MOV A,M
50AC  3D          DCR A
50AD  F5          PUSH PSW
50AE  23          INX H

```

50AF	CD0A51	CALL	TSTREM
50B2	7E	MOV A,M	
50B3	12	STAX D	
50B4	13	INX D	
50B5	70	MOV M,B	
50B6	23	INX H	
50B7	7E	MOV A,M	
50B8	12	STAX D	
50B9	13	INX D	
50BA	71	MOV M,C	
50BB	78	MOV A,B	
50BC	12	STAX D	
50BD	13	INX D	
50BE	79	MOV A,C	
50BF	12	STAX D	
50C0	13	INX D	
50C1	EB	XCHG	
50C2	221254	SHLD	PTABLE
50C5	EB	XCHG	
50C6	F1	POP PSW	
50C7	E5	PUSH H	
50C8	60	MOV H,B	
50C9	69	MOV L,C	
50CA	EB	XCHG	
50CB	2A0E54	LHLD	IN
50CE	19	DAD D	
50CF	44	MOV B,H	
50D0	4D	MOV C,L	
50D1	E1	POP H	
50D2	D1	POP D	
50D3	CD30DE	CALL	0DE30H
50D6	CD14DE	CALL	0DE14H
50D9	DAA550	JC	REN010
50DC	2A9F02 REN2	LHLD	29FH
50DF	EB	XCHG	
50E0	2AA102	LHLD	2A1H
50E3	2B	DCX H	
50E4	C5	PUSH B	
50E5	42	MOV B,D	
50E6	4B	MOV C,E	
50E7	221801	SHLD	118H
50EA	EB	XCHG	
50EB	221901	SHLD	119H
50EE	60 RLS20	MOV H,B	
50EF	69	MOV L,C	
50F0	CD14DE	CALL	0DE14H
50F3	CAFC50	JZ	RLSFIN
50F6	CD9451	CALL	SLINE
50F9	C3EE50	JMP	RLS20
50FC	B7 RLSFIN	ORA A	
50FD	C1	POP B	
50FE	3A0D54	LDA	RESERV
5101	CD08D8	CALL	0D808H
5104	E1 EXIT	POP H	
5105	D1	POP D	
5106	C1	POP B	
5107	F1	POP PSW	
5108	FB	EI	

5109	C9	RET	
510A	F5 TSTREM	PUSH PSW	
510B	E5	PUSH H	
510C	D5	PUSH D	
510D	C5	PUSH B	
510E	23	INX H	
510F	23	INX H	
5110	7E	MOV A,M	
5111	FEA9	CPI	0A9H
5113	C22851	JNZ	EXITR
5116	23	INX H	
5117	7E	MOV A,M	
5118	B7	ORA A	
5119	CA2851	JZ	EXITR
511C	CD2D51	CALL	EXVAL
511F	E1	POP H	
5120	CD14DE	CALL	0DE14H
5123	D22751	JNC	EXITR2
5126	EB	XCHG	
5127	E5 EXITR2	PUSH H	
5128	C1 EXITR	POP B	
5129	D1	POP D	
512A	E1	POP H	
512B	F1	POP PSW	
512C	C9	RET	
512D	E5 EXVAL	PUSH H	
512E	0600	MVI B	0H
5130	3C	INR A	
5131	4F	MOV C,A	
5132	23 EXVAL3	INX H	
5133	04	INR B	
5134	7B	MOV A,B	
5135	FE06	CPI	6H
5137	D24B51	JNC	EXVAL2
513A	B9	CMF C	
513B	D24B51	JNC	EXVAL2
513E	7E	MOV A,M	
513F	E67F	ANI	7FH
5141	FE30	CPI	30H
5143	DA4B51	JC	EXVAL2
5146	FE3A	CPI	3AH
5148	DA3251	JC	EXVAL3
514B	05 EXVAL2	DCR B	
514C	CA8F51	JZ	EXIVA
514F	04	INR B	
5150	210000	LXI H	0H
5153	221454	SHLD	MVALN
5156	E1 EXVAL6	POP H	
5157	23 EXVAL5	INX H	
5158	05	DCR B	
5159	CA8A51	JZ	EXVAL4
515C	7E	MOV A,M	
515D	E67F	ANI	7FH
515F	D630	SUI	30H
5161	CA5751	JZ	EXVAL5
5164	E5	PUSH H	
5165	11D852	LXI D	TABDEC
5168	2600	MVI H	0H

SPL V1.1 PAGE 5 RENUMBER WITH REMTEST (DAICLIC NO 15)

516A	68	MOV L,B	
516B	2B	DCX H	
516C	29	DAD H	
516D	19	DAD D	
516E	5E	MOV E,M	
516F	23	INX H	
5170	56	MOV D,M	
5171	210000	LXI H	0H
5174	19	EXVAL7 DAD D	
5175	DA8F51	JC	EXIVA
517B	3D	DCR A	
5179	C27451	JNZ	EXVAL7
517C	EB	XCHG	
517D	2A1454	LHLD	MVALN
5180	19	DAD D	
5181	DA8F51	JC	EXIVA
5184	221454	SHLD	MVALN
5187	C35651	JMP	EXVAL6
518A	2A1454	EXVAL4 LHLD	MVALN
518D	EB	XCHG	
518E	C9	RET	
518F	E1	EXIVA POP H	
5190	110000	LXI D	0H
5193	C9	RET	
5194	D5	SLINE PUSH D	
5195	E5	PUSH H	
5196	03	INX B	
5197	0A	LDAX B	
5198	320001	STA	100H
519B	03	INX B	
519C	0A	LDAX B	
519D	320101	STA	101H
51A0	0B	DCX B	
51A1	CDAEEF	CALL	0EFAEH
51A4	3E08	MVI A	8H
51A6	CD2ADB	CALL	0DB2AH
51A9	CDBF51	L0E262 CALL	STAT
51AC	0A	LDAX B	
51AD	B7	ORA A	
51AE	F2B851	JP	L0E263
51B1	CDF5EF	CALL	0EFF5H
51B4	3A	DB	3AH
51B5	C3A951	JMP	L0E262
51B8	CDF5EF	L0E263 CALL	0EFF5H
51BB	0D	DB	0DH
51BC	E1	POP H	
51BD	D1	POP D	
51BE	C9	RET	
51BF	0A	STAT LDAX B	
51C0	03	INX B	
51C1	FE89	CPI	89H
51C3	CAF151	JZ	GOTO
51C6	FE8A	CPI	8AH
51C8	CAF151	JZ	GOTO
51CB	FEA8	CPI	0A8H
51CD	CA0752	JZ	IFTHEN
51D0	FEA7	CPI	0A7H
51D2	CA1952	JZ	IFGOTO

SPL V1.1 PAGE 6

RENUMBER WITH REMTEST (DAICLIC NO 15)

51D5	FE94	CPI	94H
51D7	CAF151	JZ	GOTO
51DA	FE95	CPI	95H
51DC	CA3D52	JZ	LIST
51DF	FEAE	CPI	0AEH
51E1	CA5452	JZ	ONGOTO
51E4	FEAF	CPI	0AFH
51E6	CA7552	JZ	ONGOS
51E9	FEA6	CPI	0A6H
51EB	CA2752	JZ	IFSTAT
51EE	C3CEEC	STAT0 JMP	0ECCEH
51F1	CDF751	GOTO CALL	NLINE
51F4	C3CEEC	JMP	0ECCEH
51F7	F5	NLINE PUSH PSW	
51F8	0A	LDAX B	
51F9	03	INX B	
51FA	57	MOV D,A	
51FB	0A	LDAX B	
51FC	5F	MOV E,A	
51FD	CD8652	CALL	FIND
5200	7B	MOV A,E	
5201	02	STAX B	
5202	0B	DCX B	
5203	7A	MOV A,D	
5204	02	STAX B	
5205	F1	POP PSW	
5206	C9	RET	
5207	CD3652	IFTHEN CALL	PRIF
520A	CDFCEF	CALL	0EFFCH
520D	CD78CE	CALL	0CE78H
5210	15EE	DW	0EE15H
5212	CDF751	IFL CALL	NLINE
5215	CDAEEF	CALL	0EFAEH
5218	C9	RET	
5219	CD3652	IFGOTO CALL	PRIF
521C	CDFCEF	CALL	0EFFCH
521F	CD78CE	CALL	0CE78H
5222	F9CB	DW	0CBF9H
5224	C31252	JMP	IFL
5227	CD3652	IFSTAT CALL	PRIF
522A	CDFCEF	CALL	0EFFCH
522D	CD78CE	CALL	0CE78H
5230	15EE	DW	0EE15H
5232	03	INX B	
5233	C3BF51	JMP	STAT
5236	21E252	PRIF LXI H	IF
5239	CDD4DA	CALL	0DAD4H
523C	C9	RET	
523D	21E652	LIST LXI H	SLIST
5240	CDD4DA	CALL	0DAD4H
5243	CDF751	CALL	NLINE
5246	CDAEEF	CALL	0EFAEH
5249	CDF5EF	CALL	0EFF5H
524C	2D	DB	'-'
524D	CDF751	CALL	NLINE
5250	CDAEEF	CALL	0EFAEH
5253	C9	RET	
5254	21EC52	ONGOTO LXI H	SON

SPL V1.1 PAGE 7 RENUMBER WITH REMTEST (DAICLIC NO 15)

```

5257 CDD4DA CALL 0DAD4H
525A CDFCEF CALL 0EFFCH
525D CD78CE CALL 0CE78H
5260 F9CB DW 0CBF9H
5262 0A ON00 LDAX B
5263 03 INX B
5264 F5 PUSH PSW
5265 CDF751 ON10 CALL NLINE
5268 CDAEEF CALL 0EFAEH
526B F1 POP PSW
526C 3D DCR A
526D C8 RZ
526E F5 PUSH PSW
526F CD70CE CALL 0CE70H
5272 C36552 JMP ON10
5275 21EC52 ONGOS LXI H SON
5278 CDD4DA CALL 0DAD4H
527B CDFCEF CALL 0EFFCH
527E CD78CE CALL 0CE78H
5281 01CC DW 0CC01H
5283 C36252 JMP ON00
5286 7A FIND MOV A,D
5287 B3 ORA E
5288 C8 RZ
5289 C5 PUSH B
528A D5 PUSH D
528B 42 MOV B,D
528C 4B MOV C,E
528D 2A1254 LHLD PTABLE
5290 EB XCHG
5291 2AA302 LHLD 2A3H
5294 78 FIND10 MOV A,B
5295 BE CMP M
5296 23 INX H
5297 C2A752 JNZ FIND20
529A 79 MOV A,C
529B BE CMP M
529C C2A752 JNZ FIND20
529F 23 OK INX H
52A0 D1 POP D
52A1 56 MOV D,M
52A2 23 INX H
52A3 5E MOV E,M
52A4 C3D652 JMP FFIND
52A7 23 FIND20 INX H
52A8 23 INX H
52A9 23 INX H
52AA CD14DE CALL 0DE14H
52AD DA9452 JC FIND10
52B0 CD5EDD CALL 0DD5EH
52B3 F5 PUSH PSW
52B4 3E00 MVI A 0H
52B6 323101 STA 131H
52B9 21F052 LXI H MESS
52BC CDD4DA CALL 0DAD4H
52BF C5 PUSH B
52C0 010001 LXI B 100H
52C3 CDAEEF CALL 0EFAEH

```

SPL V1.1 PAGE 8 RENUMBER WITH REMTEST (DAICLIC NO 15)

```

52C6 C1 POP B
52C7 CD5EDD CALL 0DD5EH
52CA 3E01 MVI A 1H
52CC 323101 STA 131H
52CF F1 POP PSW
52D0 3E08 MVI A 8H
52D2 CD2ADB CALL 0DB2AH
52D5 D1 POP D
52D6 C1 FFIND POP B
52D7 C9 RET
52D8 0100 TABDEC DW 1H
52DA 0A00 DW 0AH
52DC 6400 DW 64H
52DE E803 DW 3E8H
52E0 1027 DW 2710H
52E2 494620 IF DB 'IF '
52E5 00 DB 0H
52E6 4C4953 SLIST DB 'LIST
52EB 00 DB 0H
52EC 4F4E20 SON DB 'ON
52EF 00 DB 0H
52F0 425241 MESS DB 'BRANCHEMENT INEXISTANT EN LIGNE
5311 00 DB 0H
5312 454E43 COMM1 DB 'ENLENCHEZ (AU CHOIX) L'
5329 27 DB 27H
532A 494D50 DB 'IMPRIMANTE'
5334 0D0D DB 0DH,0DH
5336 505245 DB 'PRESSEZ <SPACE> POUR RENUMEROTER'
5356 0D0D DB 0DH,0DH
5358 3C4252 DB '<BREAK> POUR SORTIR'
536B 0D0D00 DB 0DH,0DH,0H
536E 464952 COMM2 DB 'FIRST=0, RENUMEROTATION IMPOSSIBLE'
5390 0D0D00 DB 0DH,0DH,0H
5393 494E43 COMM3 DB 'INCREMENT=0, RENUMEROTATION IMPOSSIBLE
53B9 0D0D00 DB 0DH,0DH,0H
53BC 535441 COMM4 DB 'STATEMENTS AU-DESSUS DE.65535 APRES RENUMEROTA
ON, '
53EF 0D DB 0DH
53F0 205245 DB ' RENUMEROTATION IMPOSSIBLE'
540A 0D0D00 DB 0DH,0DH,0H
540D 00 RESERV NOP
540E 00 IN DS 2H
5410 00 FIRST DS 2H
5412 00 PTABLE DS 2H
5414 00 MVALN DS 2H
5416 00 END END

```

```

RENUM :5000 SUITE1:5041 SUITE2:5057 TESTL :505A TEST10:5068 TEST20:5084
REN1 :5091 REN010:50A5 REN2 :50DC RLS20 :50EE RLSFIN:50FC EXIT :5104
TSTREM:510A EXITR2:5127 EXITR :5128 EXVAL :512D EXVAL3:5132 EXVAL2:514B
EXVAL6:5156 EXVAL5:5157 EXVAL7:5174 EXVAL4:518A EXIVA :518F SLINE :5194
L0E262:51A9 L0E263:51B8 STAT :51BF STAT0 :51EE GOTO :51F1 NLINE :51F7
IFTHEN:5207 IFL :5212 IFGOTO:5219 IFSTAT:5227 PRIF :5236 LIST :523D
ONGOTO:5254 ON00 :5262 ON10 :5265 ONGOS :5275 FIND :5286 FIND10:5294
OK :5295 FIND20:52A7 FFIND :52A4 TABDEC:52D8 IF :52E2 SLIST :52E6
SON :52EC MESS :52F0 COMM1 :5312 COMM2 :536E COMM3 :5393 COMM4 :53BC
RESERV:540D IN :540E FIRST :5410 PTABLE:5412 MVALN :5414 END :5416

```

REMARQUEZ VOS REMARQUES EN SPL

Claudius Morin, 6 rue robert Houdin 14000 CAEN ☎ 31851532

Vous pouvez mettre en valeur vos remarques dans un programme en assembleur SPL en utilisant le petit programme suivant.

La seule obligation, c'est d'écrire le texte des remarques après la 2*** tabulation.

Le programme teste cette 2*** tabulation et s'il n'y en a pas une 3***, il remplace le code 09H de la 2*** tabulation par le code 0EH qui est caractéristique de l'écriture expansée sur une ligne pour une imprimante au standard EPSON.

Utilisation :

chargez le langage SPL, puis le code source contenant des remarques, vérifiez que le texte commence à la 2*** tabulation, repassez en UTILITY, chargez et lancez la routine, revenez au SPL, votre code source est modifié et peut être imprimé.

SPL V1.1 PAGE 1

```

1 ;
2 ;
3 ; Expansé sur une ligne
4 ; de remarque SPL
5 ; sur imprimante Citizen 120D
6 ; en remplaçant le 2eme TAB (09)
7 ; par le code 14D
8 ;
9 ; Claudius Morin
10 ; juillet 1988
11 ; 31 85 15 32
12 DEBSOU EQU 8503H ;contient l'adresse du debut du source SPL
13 FINSOU EQU 8505H ; " " de fin
14 ;
15 CPHLDE EQU 0DE14H ;compare HL et DE
16 ADHLA EQU 0DE30H ;HL=HL+A
17 ;
18 REM EQU 0FDH ;code ligne de remarque
19 TAB EQU 9H ;code de tabulation
20 EXP EQU 0EH ;code ligne expansée sur imprimante
21 ;
22 ;
23 ORG 300H
24 ;
25 PUSH B
26 CALL DEB
27 POP B
28 RET
29 ;
30 DEB LHL DEBSOU
31 XCHG FINSOU
32 LHL DEBSOU
33 BCL1 CALL CPHLDE
34 RNC
35 ;
36 MOV A,M
37 ANI 3FH
38 MOV B,A ;longueur
39 INX H
40 PUSH H
41 CALL ADHLA
42 XTHL
43 MOV A,M
44 CPI REM
45 JNZ BCL2 ;suite si remarque
46 MOV A,B
47 CPI 3H
48 JC BCL2 ;suite si + de 3 octets
49 INX H
50 MOV A,M
51 CPI TAB
52 JNZ BCL2 ;suite si 1er tab
53 INX H
54 MOV A,M
55 CPI TAB
56 JNZ BCL2 ;suite si 2eme tab
57 INX H
58 MOV A,M
59 CPI TAB
60 DCX H
61 JZ BCL2 ;saut si 3eme tab
62 MVI A,EXP ;expansé sur une ligne
63 MOV M,A
64 BCL2 POP H
65 JMP BCL1
66 ;
67 FIN END

```

GRAPHIC-TOOLKIT III : figures géométriques.

CIRCLE x,y rayon couleur

CIRCLE dessine un cercle dans n'importe quel mode graphique.

- x,y désigne le centre du cercle :
x = 0 à 511
y = 0 à 255
- rayon donne le rayon du cercle, ce dernier peut dépasser les limites de l'écran sans donner d'erreur (la partie hors écran sera ignorée).
- couleur peut prendre les valeurs 0 à 23.

ELLIPSE x,y rayon-x,rayon-y couleur

ELLIPSE dessine une ellipse dans n'importe quel mode graphique.

- x,y désigne le centre de l'ellipse :
x = 0 à 511
y = 0 à 255
- rayon-x,rayon-y donne la distance en X et Y de l'ellipse, cette dernière peut dépasser les limites de l'écran sans donner d'erreur (la partie hors écran sera ignorée).
- couleur peut prendre les valeurs 0 à 23.

remarque :

- 1) ELLIPSE est le seul moyen d'obtenir un cercle en mode 7/8 :
pour avoir dans ce mode CIRCLE X,Y R C
il faut donner ELLIPSE X,Y R*1.5,R C
- 2) de même pour obtenir un carré, la coordonnée X doit avoir un facteur de correction de 1.5

FILL=ON
FILL=OFF

Cette instruction permet de remplir (FILL=ON) ou ne pas remplir (FILL=OFF) le cercle ou l'ellipse avec la couleur donnée dans l'instruction CIRCLE ou ELLIPSE.
Après RUN, on a FILL=OFF.

FRAME x1,y1 x2,y2 couleur
FRAME x1,y1 x2,y2 couleur épaisseur
FRAME x1,y1 x2,y2 couleur épaisseur-x,épaisseur-y

Avec cette instruction, on peut dessiner des cadres de différentes épaisseurs.

Dans le premier cas, on dessine un rectangle d'épaisseur 1 (=1 ligne) sans remplir la figure. Cela fait gagner 4 DRAW.

Dans le deuxième cas, on dessine un rectangle d'épaisseur quelconque le cadre se remplit vers l'intérieur si épaisseur > 1.

Dans le troisième cas, on dessine un rectangle d'épaisseur différente en X et en Y.

POLY x1,y1 x2,y2 xn,yn couleur

Avec cette instruction, on peut dessiner une ligne brisée fermée : par exemple un polygone.

De la première paire de coordonnées, on trace une ligne jusqu'à la deuxième, etc jusqu'à la paire de coordonnées xn,yn et pour fermer de xn,yn à x1,y1.

MDRAW x1,y1 x2,y2 xn,yn couleur

Avec cette instruction, on peut dessiner une ligne brisée non fermée

De la première paire de coordonnées, on trace une ligne jusqu'à la deuxième, etc jusqu'à la paire de coordonnées xn,yn.

GRAPHIC-TOOLKIT III : remplissage de surfaces.

Ce toolkit offre quelques instructions intéressantes pour le remplissage de figures bordées dans tous les modes graphiques 4 couleurs. Il est aussi possible de remplir plusieurs surfaces en même temps. Ceci nécessite une place mémoire supplémentaire.
La coordonnée x,y donnera l'endroit où on démarre le remplissage avec la couleur donnée par FCOLOR jusqu'au bord dont la couleur est donnée par MCOLOR (ces 2 couleurs peuvent être identiques).
Ces paramètres seront mémorisés dans le buffer par STORE. PAINT commencera le remplissage des figures selon les renseignements du buffer. Ce dernier doit être vide au départ, on peut le vider par l'instruction EMPTY ou RUN. Après un PAINT, le buffer est vidé.
Le remplissage de figures peut être stoppé par BREAK.
Si on donne un ordre PAINT avant le STORE ou EMPTY suivant, s'il n'y a pas de changement de mode (excepté changement de mode x <-> xA), le remplissage sera effectué jusqu'à la fin.

FCOLOR=couleur

La couleur donnée par FCOLOR sera la couleur de remplissage et sera donc implicite pour STORE ou PAINT si elle n'est pas donnée directement dans une de ces deux instructions STORE ou PAINT.
couleur peut prendre les valeurs 0 à 23.
A l'initialisation FCOLOR=22.

MCOLOR=couleur

La couleur donnée par MCOLOR sera la couleur de bord des figures.
Remarquons que 0/XMAX/YMAX seront considérés comme bords.
couleur peut prendre les valeurs 0 à 23.
A l'initialisation MCOLOR=21.

STORE x,y cr cb
STORE x,y cr
STORE x,y

cr = couleur de remplissage.
cb = couleur de bord.

Les différents paramètres seront mémorisés dans le buffer, paramètres avec lesquels on commencera le remplissage des figures lors du PAINT. Si on veut mettre plus d'informations que le buffer ne peut en contenir, ces dernières seront perdues.
Les valeurs des différents paramètres seront validées lors du PAINT.

x = 0 à 511
y = 0 à 255
cr = 0 à 23
cb = 0 à 23

PAINT x,y cr cb
PAINT x,y cr
PAINT x,y

cr = couleur de remplissage.
cb = couleur de bord.

Si après l'instruction PAINT, on ne donne pas de paramètres, on démarre le remplissage des figures selon les paramètres stockés dans le buffer.
Si on donne des paramètres, cela correspond à un STORE avant le PAINT (cela permet par la seule instruction PAINT de mémoriser les dernières données).

x = 0 à 511
y = 0 à 255
cr = 0 à 23
cb = 0 à 23

EMPTY

Cette instruction permet de vider le buffer.

RESERVE n
RESERVE adresse-départ,adresse-fin
ST.BUF= adresse

Comme dit précédemment, on a besoin d'un buffer pour stocker les coordonnées et les couleurs des figures à remplir.

Dans le premier cas, n donne la taille du buffer et ST.BUF= donne l'adresse de début du buffer (en standard ST.BUF=#2EC).

Dans le deuxième cas, on n'a plus besoin de ST.BUF=
Pour mémoriser un ordre, on a besoin de 5 bytes. On peut donc calculer facilement la taille du buffer.

Exemple : si on veut mémoriser 64 ordres, on aura besoin de 64 * 5 bytes = 320 bytes (= #140)
ST.BUF= #2EC
buffer= #140
HEAP = #42C

Remarques :

1. Si on ne définit pas de buffer, on aura le message : UNDEFINED STORE-BUFFER
2. On peut définir le buffer dans un tableau en BASIC :
DIM A(255,1):RESERVE VARPTR(A(0,0)),VARPTR(A(255,1))+3
3. Il existe encore une limite pour PAINT au niveau de la couleur de remplissage : la couleur de remplissage, par exemple, COLOR=23 sera reconnue erronément comme couleur de bord lorsque la couleur de remplissage se trouvera strictement horizontale entre deux points de couleur de bord. Par contre si la ligne n'est pas strictement horizontale, la couleur de remplissage sera exécutée normalement.

CPAINT x,y cr cb 1,..., cb n

cr = couleur de remplissage.
cb 1,...,cb n = couleurs de bord.

Cette version de PAINT travaille aussi en mode 16 couleurs.
On peut donner ici plusieurs couleurs de bord mais la couleur de remplissage ne peut être une des couleurs de bord.
Le remplissage commencera aux coordonnées x,y avec la couleur de remplissage cr.
CPAINT n'a pas besoin de buffer mais on ne remplit qu'une figure à la fois.
0/XMAX/YMAX sont considérés comme bords.
En mode 16 couleurs, on peut utiliser les couleurs 20 à 23, on utilisera les couleurs correspondantes du COLORG.

x = 0 à 511
y = 0 à 255
cr = 0 à 23
cb = 0 à 23

GRAPHIC-TOOLKIT III : hardcopy.
=====

Ce hardcopy peut travailler dans tous les modes graphiques avec des facteurs d'agrandissement et nuances de gris quelconques.
Le choix du type de résolution, du nombre de nuances de gris et de la couleur d'écran représentée par une nuance de gris, se fera par l'instruction HCMODE=.
On utilisera l'instruction HCOLOR si on veut changer l'attribution standard d'une couleur d'écran à une nuance de gris.

En standard sont définies 2 tables. Dans chaque table, il y a 5 possibilités d'impression pour les imprimantes type EPSON.

HCMODE=n

n peut prendre les valeurs 0 à 9.

Il existe 2 possibilités de présentation :

- présentation droite = valeurs paires.
- orientation à 90 degrés = valeurs impaires.

Table	résolution	pts/page	nuances de gris	présentation normale	présentation à 90 degrés
0	1 x 1	480/864	2	HCMODE=0	HCMODE=1
1	1 x 1	960/864	2	HCMODE=2	HCMODE=3
2	2 x 1	480/864	3	HCMODE=4	HCMODE=5
3	3 x 2	320/432	5	HCMODE=6	HCMODE=7
4	4 x 2	240/432	9	HCMODE=8	HCMODE=9

- à l'initialisation ou après un RUN : HCMODE=0
- il existe une nuance de gris standard pour chaque couleur d'écran on pourra changer ce mode standard par l'instruction suivante.

HCOLOR couleur d'écran, nuance de gris

couleur d'écran peut prendre les valeurs 0 à 23.
nuances de gris, comme les tables, sont numérotées de 0 à nombre de nuances de gris-1.

Exemple : HCMODE=6 : les numéros des nuances de gris seront 0 à 4

La nuance de gris = 0 est le noir et cela va en augmentant jusqu'au blanc.

HCOPY Xgauche,Yhaut Xdroite,Ybas bord gauche ;

bord gauche est une valeur qui représente, exprimée en nombre de points graphiques, la marge à gauche du hardcopy.
Le nombre de colonnes est indépendant de l'agrandissement par HCMODE mais est dépendant de la résolution de l'imprimante en largeur.
Le bord gauche sera raccourci si le dessin ne tient pas sur la feuille de papier. On pourra donc centrer un dessin sur la droite en donnant une valeur très grande à bord gauche (par exemple 1000).
Si on demande à imprimer plusieurs blocs graphiques dans une même instruction HCOPIY, ces blocs seront placés les uns à côté des autres. Dans ce dernier cas, le paramètre bord gauche signifie la distance par rapport au bloc voisin.

Si l'image est plus large que la largeur du papier, on aura le message NUMBER OUT OF RANGE.

Les paires de coordonnées X,Y donne la zone d'écran à imprimer.
La première paire de coordonnées donne le premier point à imprimer en haut à gauche sur le papier et la deuxième le dernier point en bas à droite.

Si on change l'ordre des coordonnées, la présentation sur papier sera différente de celle de l'écran:

* présentation droite (HCMODE=valeurs paires)

- présentation normale	Xgauche,Yhaut	Ydroite,Ybas
- présentation à l'envers	Xdroite,Ybas	Ygauche,Yhaut
- refléter suivant l'axe des X	Xgauche,Ybas	Xdroite,Yhaut
- refléter suivant l'axe des y	Xdroite,Yhaut	Xgauche,Ybas

* orientation à 90 degrés (HCMODE=valeurs impaires)

- 90 degrés vers la droite	Xgauche,Ybas	Xdroite,Yhaut
- 90 degrés vers la gauche	Xdroite,Yhaut	Xgauche,Ybas
- rotation à gauche et reflet suivant l'axe des Y	Xgauche,Yhaut	Xdroite,Ybas
- rotation à droite et reflet suivant l'axe des Y	Xdroite,Ybas	Xgauche,Yhaut

HC.EXT=adresse

Cette instruction permet de construire soi-même ses propres tables pour le hardcopy.
Cela semble très compliqué, mais est-ce vraiment utile ?

GRAPHIC-TOOLKIT III : format d'impression.

FPRINT titre\$ bord-gauche, largeur no-page ligne/page insert outadr

FPRINT peut être abrégé en F?
HELP.F? vient en aide pour rappeler les différents paramètres.
Tous les paramètres sont optionnels (de la droite vers la gauche).
Les valeurs par défaut sont : FPRINT " " 6,76 1 66 6 #O0DD94

- titre\$: est un string qui sera imprimé au début de chaque page et suivi d'une ligne blanche. Si on a besoin d'une deuxième ligne de titre ou de plusieurs lignes blanches, on peut placer un CHR\$(13) dans le string; ce qui suivra le CHR\$(13) sera imprimé sur la ligne suivante.
Le string aura un maximum de 90 caractères.
Il existe un caractère particulier pour le titre : &
A l'endroit dans le string où on trouve ce caractère &, ce dernier sera remplacé par le numéro de page.
- bord-gauche : nombre de blancs à gauche avant d'imprimer une ligne
- largeur : nombre de caractères par ligne (y compris le nombre de caractères du bord-gauche).
- no-page : numéro de la première page (servira pour le & dans le titre).
- ligne/page : nombre de lignes par page (y compris la ligne titre et la ligne blanche).
- insert : lors d'une continuation à la ligne suivante, on donne le nombre de blancs à insérer en plus du bord-gauche (comme dans LIST du BASIC).
- adrout : normalement, l'édition est envoyée sur RS232 (#DD94) mais on peut spécifier une autre adresse pour l'édition.

Exemple: FPRINT "EXEMPLE - Page &" 10,50 10 66 6

```

-----
          1         2         3         4         5
colonnes -----> 1234567890123456789012345678901234567890
                    EXEMPLE - Page 10
                    !
                    ! 10000 FILL 0,0 100,200 21:DRAW 20,25 90,
                    !           90 5:END
                    !
                    !
                    ! *-----> insert=6
                    ! *-----> bord-gauche=10

```

FPRINT donne seulement les paramètres, pour activer l'édition, il faut taper POKE#131,8:LIST ---> écran + imprimante
POKE#131,9:LIST ---> imprimante uniquement

GRAPHIC-TOOLKIT III : activer SGT.

SGT.

Cette instruction permet de charger la partie SGT en RAM et d'initialiser le SGT. Ce dernier occupe les adresses #2EC à #F93 comprenant outre la routine, la table standard de caractères (96 signes) et le jeu de cartes.

En mode direct après avoir tapé SGT, on a le message SGT ON LINE.

On peut également utiliser SGT dans un programme BASIC mais comme toute première instruction (le programme BASIC sera décalé pour placer le SGT en RAM).

Si on veut lister un programme contenant des instructions SGT et que l'ordre SGT n'a pas été introduit, on obtient le message :
*** UNDEFINED CANAL 3

Si on veut enlever la partie SGT en RAM, on tape CONCAT 3,0 ou \$RESET
Le heap reste à la valeur #F94, on peut récupérer cette place par HEAP #2EC.

GRAPHIC-TOOLKIT III : divers.

MODE=n

Cette instruction est identique MODE du BASIC, mais ici n peut être une expression arithmétique, le résultat donnera le mode correspondant suivant :

```

ancien MODE : 1 1A 2 2A 3 3A 4 4A 5 5A 6 6A 7 7A 8 8A 0
nouveau MODE: 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 .. 255

```

Toute valeur n>15 donne un MODE 0.

CLEAN

Cette instruction efface la partie graphique de l'écran (=CLS du mode texte) et est plus rapide que FILL 0,0 XMAX,YMAX 20

Cette instruction doit être précédée d'une instruction MODE.
Pour avoir un écran propre : MODE n:CLEAN en début de programme.

CLEAN n

L'écran passe en MODE n avant de faire un CLEAN.

LES PROGRAMMES "CRUNCH".

"Crunch" permet de rassembler en un seul bloc les fichiers et programmes qui se trouvent sur une cassette et de les charger en mémoire s'il y a assez de place.

\$CRUNCH n

n donne le nombre de fichiers ou de programmes à lire.
Si n=0 on lira la bande jusqu'à la fin.

\$CRCONT

Cette instruction permet de continuer le "Crunch" après un arrêt (fin de bande, écriture sur disque après n programmes,...).

\$CRSAVE

Cette instruction donne la liste des programmes mémorisés qui seront à ne pas recopier sur disque.

N.B. : "Crunch" ne travaille que sur cassette et pas sur disque.

LE PROGRAMME "ALLADER"

"ALLADER" permet de lire automatiquement une cassette et de recopier sur disque, sous le même nom, les fichiers ou programmes.

CONCATENATION DE PROGRAMME BASIC ET MACHINE.

=====

Cette instruction intègre en un programme BASIC la partie langage machine et la partie BASIC. Lors du dépackage (RUN du programme), le langage machine (MP) sera détaché du programme BASIC et placé à son adresse correcte en mémoire centrale.

Ceci remplace donc le "Dainamic Bootstrap Loader" qui ne fonctionne pas sur disque.

1) Syntaxe :

```
$CONC:FILL o,e m,a h  (:CLEAR x)(:MODE n)(:TRON)(:LIST)
$CONC:OUT o,a         (:CLEAR x)(:MODE n)(:TRON)(:LIST)
$CONC:LOAD           (:CLEAR x)(:MODE n)(:TRON)(:LIST)
```

2) Signification des paramètres avec FILL :

o,e = adresse début, adresse de fin de MP (où la placer après RUN)
 m = adresse momentanée en RAM de MP càd si m différent de o
 a = adresse autostart de MP (a=0 si pas d'autostart)
 h = adresse de start du HEAP (si h=0 --> HEAP=e+1)

3) Signification des paramètres avec OUT :

o = adresse début de MP
 a = adresse autostart de MP (a=0 si pas d'autostart)

Avec OUT, l'adresse de fin de MP = DEEK(#29B - 1) et le MP se trouve à sa bonne place en RAM juste en dessous du HEAP.

4) Signification du paramètre LOAD :

Avec LOAD, on peut copier d'une cassette vers une disquette un programme DBL vers la méthode \$CONC.

5) Signification des paramètres de "dépackage" :

CLEAR x : la longueur x du HEAP est mémorisée, lors du dépackage un CLEAR x sera exécuté.

MODE n : avant le dépackage, une instruction MODE n est exécutée il est donc possible à la place d'un MP de dépacker des images (MODE 1 à 4A) par le RUN du BASIC.
 On peut penser à l'affichage de LOGO qui, grâce à l'option autostart a=#D6DA, reste à l'écran tant que l'on n'appuie pas sur la barre d'espacement.

TRON : après le dépackage, le MP sera séparé du programme BASIC, il sera parfois intéressant de le dépacker à chaque RUN (pour un LOGO). Avec cette option, il reste intégré au programme BASIC.

LIST : il y a certains programmes BASIC avec MP qui ne permettent pas d'être listés sans avoir fait tourner le MP (on pense à certains programmes particuliers avec une partie NCU qui sans initialisation ne peuvent être listés ou édités). Avec cette option apparaîtra un message si on essaye de lister le programme.

6) Remarques :

- Avec \$CONC, sera réservée une ligne au début du programme sans numéro de ligne : CALLM PEEK(.....)
 On peut donner un numéro de ligne par EDIT mais ne pas changer la position de la ligne dans le programme BASIC.

- Si on veut lister un programme où un MP est intégré par \$CONC on aura à la fin du listing un message OVERFLOW là où commence le MP.

7) L'ordre des opérations lors du dépackage :

- 0) d'abord RUN --> CALLM PEEK(.... --> démarrage du dépackage
- 1) LXI SP,:F900
- 2) placer la valeur du début du HEAP (h) en #29B
- 3a) remplacer l'ancienne valeur du HEAP (si mémorisée par CLEAR x dans \$CONC:FILL ...)
- 3b) longueur du HEAP=4
- 4) charger la pile avec le sous-programme et le démarrer
- 5) CLEAR (pointeurs Heap et Basic à placer/déplacer)
- 6) éventuellement faire MODE n
- 7) copier le MP (dépackage)
- 8) placer la longueur du Heap à sa vraie valeur
- 9) MP et première ligne CALLM PEEK(...) à effacer (si on n'a pas utilisé l'option TRON)
- 10) un deuxième CLEAR, RESTORE
- 11) éventuellement un Autostart
- 12) GOTO à la première ligne du BASIC

8) Remarques générales :

- le MP peut être une image; dans ce cas, il faut définir la valeur h du HEAP, sinon le programme BASIC va être chargé à l'emplacement des ROM ! (e=#BFFF alors h=#C000 !)
- On peut intégrer plusieurs MP, la dernière intégrée sera la première dépackée. Si l'option TRON est utilisée, elle doit l'être dans la première MP intégrée.
- On ne pourra rien copier en pile.
- Des exemples explicites se trouvent dans le manuel allemand (p. 76 et 77).

