(7)

```
WG1 : MOVE.L  TPPI,A5
WG1:  MOVE    YTRACE,D1
WG1:  CMP     #2,D1
      BCS     HI21A   → arrêt sur bkpt
x     BEQ     HI21D   ——→ cas step en fois
      CMP     #5,D1
x     BEQ     HI21G   ——→ cas step source
      MOVE.L  YRETL,D0
      LEA     TMFOR,A0
      CMP     #3,D1
      BNE     HI21E
                        MOVE.L YRETL,D0
                        CMP.L  TMPROC,D0
                        → fin de for
                        ↓ fin de proc ?
      CMP.L   (A0),D0
      BGT     HI21D
HI21F: MOVE.L TDEBAS,A5 ← MOVE.L TVALA6,A6
      BRA     WINSTRS
HI21E: CMP.L  (A0),D0
      BGE     HI21F
      MOVE.L  TPPI,A5
HI21D: BSR    LOCALS    → BSR XESAVH
                          localise A5 et sépare {TPI TPL
HI21H: BSR    XESAV     sauvegarde l'écran
HI10: BSR     XCLRT1    vide l'écran
[HI12] LEA    HH80,A0   } "debug"
      BSR     XPTET     }
      BSR     Y10L      Sort dix lignes
      BSR     WGINIT
      LEA     HH81,A0   menu <A propos ...
[HI13] BSR    XN15
HI14: BSR     VDKEY     ← DIR CURSACT
      BSR     XKEY
      LEA     SKBP,A5   bkpts
                  .L

HI21G: CMP.L  #TK,A5
      BCS     HI21F
      CMP.L   TDHAUT,A5
      BHI     HI21F
      BRA     HI21D
              localise A5
HI21E: MOVE.L YRETL,D0
      CMP.L   TMFOR,D0
      BGE     HI21F

{MOVE.L TDEBAS,TPPI
 MOVE.L TPL,TPPL
LEA   TDEBAS,A0
MOVE.L TDEBAS,-(SP)
       (A0)+,(A0)
MOVE.L TVALA6,A6
ST     YTRACE
LEA    PB961,A5
BSR    WSPBAS
MOVE.L (SP)+,TDEBAS
MOVE.L A6,TVALA6
LEA    TDEBAS+4,A0
MOVE.L (A0),-(A0)
```

(inhibe bkpt)

HI10: MOVE #2, RESOL

RESOL1 = HI10+2

BSR HIDECM

BSR SHOWCM ⊗

CLR -(SP)
MOVE #87,-(SP)
TRAP #14
ADDQ #4,SP

⊗ Smear mode off (TT)

BSR SETCED

BSR DAMPR1

) reset

boucle by

BSR XCLR

BSR HIDECM

LEA GEX85,A2

BSR DAMGN

} dernier de débug

BSR XCLRT1 ⊗

BSR YIOL

10 lignes

LEA TDEBAS,A0

MOVE.L (A0)+,(A0)

MOVE.L TVALA6,A6

ST YTRACE

inhibe bkpoint

LEA PB961,A5

BSR WSPBAS

} B_debug

MOVE.L A6,TVALA6

LEA TDEBAS+4,A0

MOVE.L (A0),-(A0)

BSR WGINIT

BSR HIDECM

HI14: BSR VDKEY ⊗

BSR XKEY1

BSR HIDECM

LEA SKBP.L,A5

bkpt

LEA YECRA,A1

(7) MOVE.L D0,D1
SWAP D1
CMP #1,D1

~~CMP.L #$1xxxxx,D0~~

BCC ~~HI27~~ GEX10 ⊗ CMP #$215,D0 / BEQ GEX40 — F21 ≡ @
CMP.L #$214,D0 ← BEQ HI220 → CLR
BCC HI27 → err
CMP #$200,D0
BCS GEX90
TST.B D0
BEQ HI24 — (F1) ≡ CR
CMP.B #7,D0
BCC GEX88
ADD #$40,D0
BRA GEX91 → "A" à "F" ≡ (F2) à (F7)

GEX88: BEQ HI33 → (F8) ≡ "H"
CMP.B #10,D0
BCC GEX89
ADD #$42,D0 — J K / F9 F10
BRA GEX91

GEX89: BEQ HI240 ← FM ≡ Space — { CMP.B #20,D0 / BEQ GEX40 — F21 ≡ @
CMP.B #13,D0
BEQ HI40 → F20 ≡ L
~~SUB #9,D0~~ ADD #39,D0 ↓ 2 à 9 ≡ F12 à F19
BRA GEX91

GEX90: CMP #$100,D0
BCS GEX91
CMP.B #12,D0 — CLR
BEQ HI220
CMP.B #13,D0 — CR
BNE HI27 → err

(1)

```
GEX91: CMP.B  #"2",D0          ⊗
       BCS    HI22
       CMP.B  #"9",D0
       BMI    HI22
  cas  1-9 et 0
       SUB.B  #"4",D0
```

```
⊗ GEYR: MOVE.L  TPI,A0          met bkpt = A0
  HI15 : SUBQ.B  #1,D0
         BMI     HI20
         BSR     SUITA0
         BEQ     HI27           → erreur
         BRA     HI15

  HI20 : BSR     YBKP            met le bkpt A0

  HI21 : BSR     XEREM           remet l'écran
         MOVE.L  TVALA6,A6
         BRA     HI21F           → retour
```

```
GEY10:BEQ   GEY14          → bouton gauche
                           ↓ bouton droit
      TST   (A1)
      BNE   HI27           → erreur si écran
      CLR   D6
      MOVE.B D0,D6         y = D6
      BSR   WDCUR  (1)134  la ligne D6 commence par n= ?
      BNE   HI27           → non
                             oui
      SUB.L TPL,D0
      BGT   GEY12 ← BEQ HI33   → (H) si 1ère ligne
HI27R : BRA HI27
```

```
GEY14 : TST (A1)           bouton gauche
        BNE HI27R          → erreur si écran
        LEA TD000,A1
        MOVE FCURSC,D1     -TD000(A1)
        ADDQ #1,D1         80 ou 40
        MOVE D0,D2
        EXT D0             y
        MULU D0,D1
        ASR #8,D2
        ADD D2,D1
(SP) GEY15 : LEA (A1,D1.W),A0      utilisé aussi par help
GEY15A:      LEA TYPCAR,A2 ← MOVEQ #0,D2 (X)
GEY16 : CMP.L A1,A0
        BLT GEY17
        MOVE.B -(A0),D2    ← TST
        BTST #2,(A2,D2.W)     MOVE.B (A2,D2.W)
        BNE GEY16             BMI GEY16
                              BTST #1,(A2,D2.W)
                              BNE GEY16
GEY17 : ADDQ #1,A0
        CMP.B #32,(A0)     ← RTS
        BEQ GEY17             BSR GEY15
        BSR PONT
        CMP.L A1,A0
        BGE HI27R          → cligno plus loin que le curseur
            HI14
```

```
MOVE.B  (A0),D2
BTST #6,(A2,D2.W)
BEQ   HI14                    → non alphanum

LEA   TBUF,A3   ← MOVE.L TBUF,A5   (A3)
MOVEQ  #31,D0
BRA  GEY19
GEY18:MOVE.B  D2,(A3)+
GEY19:MOVE.B  (A0)+,D2
      BTST.B #6,(A2,D2.W)
      DBRA  D0,GEY18
      CLR.B  (A3)+
      BSR    YTYP             écrit le type
                              met FLTYP
      MOVEM.L (SP)+,A3/A5  ← LEA 1(A3),A0
      MOVE.B  #FLTYP,D0   FLTYP=*-1
      BMI    HI14             non imprimable
      CMP.B  #13,D0
      BCS    GEY21            ok
      CMP.B  #16,D0
      BCS    HI14             → non imprimable (commande 13,14,15)
      CMP.B  #$50,D0
      BCS    GEY21            ok
      CMP.B  #$52,D0
      BCS    HI14             → non imprimable (labels ordinaires $50,$51)
      CMP.B  #$71,D0          (CMP.B #$55,D0 / BCS GEY28)
      BCS    GEY27            ok
      CMP.B  #$74,D0
      BCS    HI14             → $71, $72, $73 (then / of / value)
      CMP.B  #$75,D0
      BEQ    HI14             → $75 (xor...)
                              ↓ $76

{ SUBQ #1,A0
{ MOVEM.L A3/A5,-(SP)/MOVEQ #1000,D0
GEY21: CMP.L A1,A0
       BGT GEY30
       MOVE.B (A0)+,(A3)+
       BRA GEY29 DBRA
GEY30: CLR.B (A3)+
GEY29: MOVE.B (A0)+,(A3)+
       DBMP.L A1,A0
       DBGT D0,GEY29
       CLR.B (A3)+
       MOVE.L TVALA6,A6
```

```
~~GEY21:MOVE.L  TVALA6,A6~~
GEY21: CMP.B  #"(", (A0)
       BNE    GEY26
       MOVEQ  #0,D3          nb de par
GEY23: CMP.L  A1,A0
       BGT    HI14
       MOVE.B (A0)+,D0  <- BEQ HI14
       MOVE.B D0,(A3)+
       ADDQ #1,D3
       CMP.B  #"(",D0
       BEQ    GEY23
       SUBQ   #1,D3
GEY25: CMP.B  #")",D0
       BNE    GEY23
       SUBQ   #1,D3
       BNE    GEY23
       CLR.B  (A3)

GEY26: ~~MOVE.L  TVALA6,A6~~  BSR PB962
                          <-BSR WGLATT   remet YTRACE=1
       MOVE.L  A5,A0
       BSR    X15
       MOVEQ  #"=",D0
       BSR    X12B
       MOVE   #$140,D7   { moveq #0,d7
       BSR    MA28         BSR GEB74 <-(9)16b
                           exécute print
       MOVE.L  A6,TVALA6  BSR MA512
       BRA    HI14
GEY27: CMP.B  #55,D0
       BCC    GEY21
GEY28: CMP.B  #32,(A0)+
       BEQ    GEY28
       CMP.B  #":";(A0)
       BEQ    HI14
       BRA    GEY21

                  CMP.B  #34,D0
                  BNE    GEY25
GEY24:            MOVE.B (A0)+,D0  <- BEQ HI14
                  MOVE.B D0,(A3)+
                  CMP.L  A1,A0
                  BGT    HI14
                  CMP.B  #34,D0
                  BNE    GEY24
```

type 52 et 54

débug

GEX85

SSold	CR	1
		2
Arret	A	3
Break	B	4
Breaks	C	5
Return	D	6
Ecran	E	7
Fois	F	8
Cycle	H	9
Run	I/J	10
Nop	K	11
SStk	SP	11
2		
3		
:		
:		
9		
Loop	L	

débug

écran débug

GEX86:

CLR	CLR
Debug	K

écran débug

```
7   HI22: CMP.B  #$20,D0
      BEQ    HI240
      LEA    HH70,A0
      BSR    RELAD0
      BNE    HI27
      LEA    YECRA,A1
      JMP    (A0)

CR    CMP.B  #6,D0
      BEQ    HI220

HI220: TST (A1)                    LR
      BEQ    HI27   <- BSR HIDECMR
      BSR    XCLR
      BRA    HI14   <- BSR XESAV

D   HI23: MOVE.L  TMPROC,YRETC
      MOVE   #3,TRACE
      BRA    HI21

CR  HI24: MOVE   #2,YTRACE          step
      BRA    HI21

Space  HI240: MOVE   #5,YTRACE      step source
      BRA    HI21

A   HI25: BRA    HD289

B 1 bkpt  HI26: TST  (A1)
      BNE    HI27
      BSR    YRBP
      BNE    HI20
HI27:
      BRA    HH190                  err

E plusieurs bkpts  HI28: TST  (A1)
      BNE    HI27
      BSR    YRBP
      BEQ    HI27
      BSR    YBKP
HI29: BSR    YRBP
      BEQ    HI21
      BSR    YBKP
      BRA    HI29
```

(7) [E] ret écran

```
HI31: ST (A1)
      BSR   XEREM   <- ret écran
      BRA   HI14
```

[F] fois s'arrête sur un bkpt au bout de n fois

```
HI32: TST (A1)
      BNE   HI27
      ~~MOVE #1,YFOIS1~~   <- CLR TFRUN
      LEA   HH82,A0        } fois>
      BSR   3BE10          <- ST TFRUN
      TST.L D0
      BEQ   HI27
      MOVE.L D0,YFOIS      <- BLE HI27
      BSR   YRBP             demande bkpt
      ~~BNE~~ HI34
      cas bkpt dans TPI, A0
```

[H]

```
HI33: MOVE.L ~~TPI~~, ~~YBKPA~~
~~HI33A: MOVE #1,YFOIS1~~
      BRA   ~~HI24~~ HI20
~~HI34: MOVE.L D0,YBKPA~~   ~~BSR YRBP~~
      BRA   HI20
```

[H] =[I]ois et CR ~~YFois=1 ytrace=2~~

H fois = 1 bkpt = idem

```
HI33: MOVE.L TPI,A0
HI34: BRA   HI20   met bkpt
```

[I]

```
HI36: ~~MOVE #1,YTRACE~~   CLR ~~TFRDG~~ YTRACE
      BRA   HI21
```

[L]

```
HI40: MOVE #4,YTRACE   TMFOR
      MOVE.L ~~TFOR~~,YRETL
      BRA   HI21
```

@ écrit la valeur de l'argument @k

après HI32

```
GEA40: TST  (A1)
       BNE  HI27          <- BSR X9
       LEA  GEA41,A0      "numéro de l'argument"
       BSR  3BE10         -> D0.L

       MOVEA TMPROCA,A3
       LEA  GEA41,A0      "pas d'argument"
       MOVE -(A3),D4
X      BEQ  GEA45         -> écrire a0
       MOVE -(A3),D0
       EXT.L D0
X      BEQ  GEA45         "
       MOVEM.L D0/D4/A3,-(SP)
       LEA  GEA42,A0      "numéro de l'argument >"
       BSR  3BE10         -> D0
       MOVE.L D0,D3
       MOVEM.L (SP)+,D0/D4/A3
       TST.L D3           <- LEA GEA43,A0
       BGT  GEA46
GEA44: BSR  PINTAE        "^arguments"

GEA45: BSR  X15N
       BRA  HI14

GEA46: CMP.L D3,D0
       BLT  GEA44
       SUB.L D3,D0
       BSR  PB840         { A3 = début
       MOVE D0,D3         { D0.L = largeur
       BRA  GEA48
GEA47: MOVE.B (A3)+,D0
       BSR  XRA           (sans mvt)
GEA48: DBRA D3,GEA47
       BRA  HI14
```