1 met résolution RESOL

```
SETCED :MOVEM.L  D0-D7/A0-A6,-(SP)
        BSR  HIDECM
        LEA  RESOL,A0
        MOVE RESOLB(PC),D1      résolution du bureau
        CMP  #2,D1
        BGE  GEU101        ⊗ → autre que 0 et 1
        AND  #1,(A0)          rectifie RESOL (0 ou 1)   } cas ST couleur
        CMP  (A0),D1
        SNE  AESOK            ← 0 résol bureau
        MOVE (A0),D1          ← -1 autre
GEU100: MOVE D1,(SP)
        ↓
      GEU102

        BRA  GEU102                    ⊗
GEU101: CMP  (A0),d1
        BLS  GEU102  GEU10
        AND  #1,(A0)
        ADD  D1,(A0)
GEU10 : MOVEQ #2,D0            ⊗
        BRA  GEU114

        IF FLAG30-1            version ST interdit
        BLE \1                 résol > 2
        LEA GC32,A0
        BSR ALERT1
        BRA WQUIT              ⊗
\1:     CMP #2,D1
        ENDIF

RESOL : D.W  0
RESOLB:D.W   0
```

☒ version spéciale pour ST en résolution quelconque :

```
CMP.B #"0", X68000
BEQ \1                → si ST, accepte la résolution
[aussi mettre NBLIGNE et NBCOL
 (ligne 9 et 10)]
```

1 Var p 1016

```
SETCED : MOVEM.L  D0-D7/A0-A6,-(SP)
         MOVE  #2,-(SP)
         RESOL = *-2
         CMP  #2,(SP)
         BLE  GEU10
GEU102 : MOVE  #2,(SP)
GEU10  : MOVE  #4,-(SP)
         TRAP  #14
         ADDQ #2,SP
         CMP  (SP),D0
         BEQ  GEU112
```

```
         BGE  GEU102
         CMP  #2,(SP)
         Resolb = *-2
         SNE  AESOK
         BRA  GEU10
```

si couleur met aesok = {0 résol du bon / -1 autre résol

sur la pile : résolution demandée

d0 = resolution actuelle

inutile

ok

```
GEU102 : MOVE D1,-(SP)
         MOVE #-1,D0
         MOVE.L  D0,-(SP)
         MOVE.L  D0,-(SP)
         MOVE  #5,-(SP)
         TRAP  #14
         ADD #10,SP
```

résolution = D1 (0 ou 1)

change résolution

```
         ADDQ #8,SP
         MOVE #2,(SP)
         TRAP #14
         MOVE.L D0,$436.W
         MOVE.L  D0,-(SP)
         MOVE.L  D0,-(SP)
```

Physbase

ADDQ #2,SP

change l'écran

CLR.B 3(SP)

```
GEU112 : MOVE.L  $436.W,-(SP)
         MOVE.L  (SP),-(SP)
         MOVE  #5,-(SP)
         trap  #14
         addq #8,SP
         MOVE.L (SP)+,D0
```

GEU114

résolution ∈ {0,1,2}

↳ ou autre sur TT

```
CMP #2,D0
BCS GEU114
MOVEQ #2,D0
```

si resol > 2 met d0=2

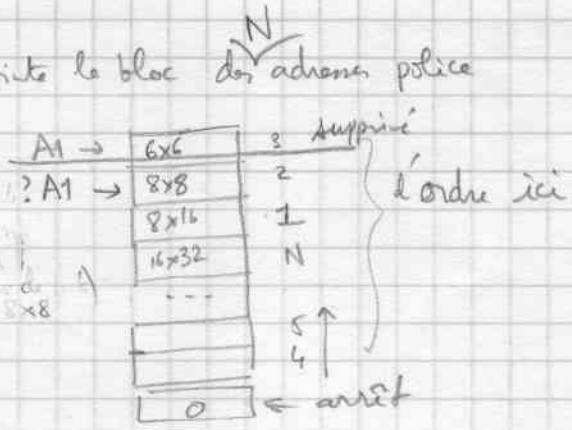
si d0 ∈ {0, 1, 2} (si resol ≥ 2)

resol		police	d4 haut2k	d2-1 tcurs6	d3-1 tcurs8	d1 width2k
0	basse	8x8	3	24	39	3
1	moy		3	24	79	3
2	haute	8x16	4	24	79	3
3	"	8x8	3	49	79	3
		16x32	5			4

```
GEU114:LEA COLRES,A3
       MULU #6,D0
       ADD D0,a3
       BSR SETCOL4
       MOVE RESOL,D0
       MOVEQ #3,d1        ← MOVEQ #3,d4
       MOVEQ #80,d3
       MOVEQ #25,d2
       CMP #1,D0
       BEQ GEU14          → résol = 1
       BCC GEU12          → résol > 1
       MOVEQ #40,d3       ↓ si résol = 0
       BRA GEU14
GEU12: MOVEQ #40,d3       screen_x
       MOVEQ #4,d1        screen_pc
       MOVEQ #16,d2       screen_c
       MOVE #$4140,d4     screen_px
       CMP #1,d0
       BHI GEU12          → resol > 1
       BNE GEU116         → = 0
       MOVEQ #80,d3       screen_x
       MOVEQ #2,d1        screen_pc
       MOVEQ #4,d2        screen_c
       MOVE #$4280,d4     screen_px
GEU116:LEA RESOLPX,A0
       MOVE d4,(A0)
       MOVE.B d1,RESOLPC-RESOLPX+1(A0)
       MOVE.B d2,RESOLI-RESOLPX+1(A0)
       MOVEQ #3,d1        width2k
       MOVEQ #25,d2       screen_y
       MOVEQ #3,d4        haut2k
       BRA GEU14
```

```
GEU12:D.W   $A000          A1 pointe le bloc des adresses police
GEU1A:MOVE.L A1,A0   <ADDQ #4,A1
GEU20:TST.L (A0)+
      BNE GEU120
      MOVE.L A0,D1
      SUB.L A1,D1
      ASR #2,D1
      SUBQ #1,D1        N
      MOVEQ #0,D0
      LEA RESOL,A2
      MOVEM (A2),D0/D4 ;read/8
SUB   D4,D0
DIVU  D1,D0
SWAP  D0
ADD   D0,D4
MOVE  D4,(A2)

LEA   8(A1),A0
CMP   #6,D4
BNE   \1
ADDQ  #4,A0
\1:
```



rectifie RESOL
de sorte que RESOL-RESOLB ∈ [0,N)

pointe police N
si haute résolution TT

```
        ADDQ #1,D0          <- GEU121: MOVE.L -(A0),A2
GEU121: SUBQ #1,D0
        BEQ  GEU126         -> fin

GEU122: MOVE.L -(A0),A2
        CMP.L A1,A0
        BGT  GEU121
GEU124: TST.L (A0)+         } avance A0 en fin de bloc
        BNE  GEU124
        SUBQ #4,A0
        BRA  GEU121
GEU126: MOVE.L #$FFFF,D1    <- LEA RESOLPX,A3   caractéristiques de la police A2
        MOVE RESOLPX(A3),D3 ; RESOLPX
        AND.L D1,D3
        DIVU $34(A2),D3     <- MOVE $34(A2),D5 / ADDQ #7,D5 / AND #$FFF8,D5 <- MOVE.B D5,RESOLCX-RESOLPX+1(A3)
        AND #$FFFC,D3       <- BSR GEU16   met D4 tel que 2^d4 = d5   ⊗
                               d3 = nb de colonnes (multiple de 4)
        MOVE RESOLPY-RESOLPX(A3),D2
        AND.L D1,D2
        MOVE $52(A2),D5     <- d5 = hauteur en pixels du caractère
        DIVU D5,D2          <- d2 = nb de lignes
                               MOVE.B D5,RESOLCY-RESOLPX+1(A3)
                               met d4 : 2^(d4-1) < d5 ≤ 2^d4
        MOVEQ #1,D6
        MOVEQ #0,D1
GEU128: CMP  D5,D6          MOVE D4,D1
        BCC  GEU130         BSR GEU16   ⊗
        ADDQ #1,D1
        ADD  D6,D6
        BRA  GEU128
GEU130: MOVE #T1173-TD000,d6   <- nb de caractères total max
        DIVU D3,d6             -> nb max de lignes de d3 car
        CMP  d2,d6
        BCC  GEU131
        MOVE d6,d2
GEU131: MOVEQ #CONTRL-T1173-4,d6   -> limité dû à T1173
        CMP  d2,d6
        BCC  GEU132
        MOVE d6,d2
             |
             v
          GEU132
```

```
GEU32:MOVEM  D1-D4,-(SP)        ⊗        initialise la police
      MOVEM.L  VDIPB, A0/A1
                              contrl intlin
      MOVE.L   A2,(A1)                   appel vdi 5/102
      MOVE     #5,(A0)+
      CLR.L    (A0)+
      MOVE     #2,(A0)+
      CLR      (A0)+
      MOVE     #102,(A0)+
      MOVE     UAESO,(A0)+
      BSR      VDI
      MOVE     (SP)+,D1-D4       ⊗
               ↓
             GEU14
```

```
GEU14:LEA  TCURSBM, A0
      LEA  RESOLX+1, A3
      MOVE.B  d3,(A3) ;resolx+1
      MOVE.B  d2, resoly-resolx(a3)
      SUBQ #1,d3
      SUBQ #1,d2
      MOVE  D4, HAUT2K  ⊗ <-- MOVEM D1/D4, WIDTH2K  ⊗
      MOVE  D2,(A0) ;Tcursbm
      MOVE  D2, TCURSB-TCURSBM(A0)
      MOVE  D3, TCURSC-TCURSBM(A0)
[GEU16] BSR  DAMPR1 )
      BSR  EMULNOR )
      MOVEM.L  (SP)+, d0-d7/a0-a6
GEU15 : RTS                    ⊗
```

remet cursb

```
if flagry-1
MOVEQ #0,D0          ⊗ n6 de ligne pour debug
MOVE  D2,D0
DIVU  #3,D0
SUB   D0,D2
MOVE  D2,TCURSBL
endif
```

```
WIDTH2K: D.W  3   1
HAUT2K : D.W  3   1
```

3 car de hauteur 8
4 " 16
5 " 32

3 car de largeur 8 ⊗
4 " 16