initialise mémoire / efface les noms externes (anciens)

(3)

```
        BSR   XMKILL        <- efface ancien nom
INITR : LEA   TMIND,A0                 BSR  IMOOREO    Randomize ⊗
        LEA   TK.L,A1       } limite   (barré)
        MOVE  #1021,D0      indexpile
PA20  : ADDQ  #4,A0
        MOVE.L A0,D1
PA21  : MOVE.L D1,A2
        MOVE.L -(A2),D1
        BEQ   PA22
        CMP.L A0,D1         ⊗ (table TMIND après en fin } sert de limite
        BCS   PA21
        CLR.L (A2)
PA22  : DBRA  D0,PA20

        MOVE.L TDHAUT,A6
        MEVEN  A6
        LEA   TMNDES,A0     pointeurs à initialiser
        LEA   TCM1,A1       longueurs
        MOVE.L (A1)+,D0     tampon source
        ADD.L D0,A6
        MOVE.L A6,(A0)+     tmndes
        MOVE  #$7F7F,(A6)+  flag de non destruction
        CLR   (A6)+         arrêt du dépilage des procédures
        MOVE.L A6,(A0)+     tmproc

        MOVE  (A1)+,D0      nb de procédures -1   (mots de la pile et for)
PA23  : CLR.L (A6)+
        ADD   #12,A6        (barré)
        DBRA  D0,PA23       <- MOVE.L A6,(A0)+ ⊗ tmfor
        MOVEQ #-1,D6        } marque d'arrêt de dépilage de for (next)
⊗     { MOVE  D6,(A6)+      (barré)
      { CLR.L (A6)+         (barré)
        MOVE  (A1)+,D0
PA24  : CLR   (A6)+
        ADD   #14,A6
        DBRA  D0,PA24
        MOVE  D6,(A6)+      -1 : marque d'arrêt d'empilage des for (barré)
```

⊗ tampsrc

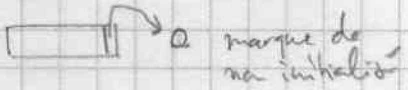
```
ADD.L  TCM1,A6        tampon source
MOVE.L A6,TMNDES
BSR    INITS
```

(3)

```
        MOVE.L   PILEEXEC,D0
        MOVE     #$1A1A,D1
        BRA      PA240
PA24 :  MOVE     D1,(A6)+
PA240:  SUBQ.L   #2,D0
        BPL      PA24
        MOVE.L   A6,(A0)+  ; tmcond
        MOVE.L   A6,(A0)+  ; tmcondp
        MOVE     PILECOND,D0      nb de conditions-1
PA25 :  CLR      (A6)+
        CLR.L    (A6)+
        DBRA     D0,PA25
        MOVE     #-1,(A6)+        ⊗ arrêt de condition
        MOVE     MEMFILES,D0      table des mém-files
        SUBQ     #1,D0
        MOVE.L   A6,(A0)+ ; TMMEM
GA22 :  CLR      (A6)+
        CLR.L    (A6)+
        DBRA     D0,GA22
        MOVE     d0,(a6)+         arrêt
repete> MOVE     RFILES,D0        table des R-files
        SUBQ     #1,D0
        MOVE.L   A6,(A0)+ ;TMRAN
GA23 :  CLR      (A6)+
        CLR.L    (A6)+
        DBRA     D0,GA23
        MOVE     d0,(a6)+         arrêt
        MOVE.L   A6,(A0)+ ;TMENU  table des menu
        MOVE.L   TAMPMENU,d0
        LSRL     #2,d0
GA230:  CLR.L    (A6)+
        DBRA     d0,GA230
        SUBQ.L   #1,d0
        BPL      GA230
```

```
        MOVE.L   A6,(A0)+  ; TMREM
        MOVE     PILEREM, D2   <--  { MOVE D2,D0
                                    { ASL  #2,D0
        MOVE     D0,(A6)+        0 [ 4n ]
        MOVEQ    #4,D0
        MOVE     D2,D1
        BRA      GAI23
GAI22:  MOVE     D0,(A6)+        0   [4]
        SUBQ     #4,D0           4   [0]
        MOVE     D0,(A6)+
        ADDQ     #8,D0           4n  [4n-1]
GAI23:  DBRA     D1,GAI22
        CLR      (A6)+           4n  [0]

        MOVE.L   A6,(A0)+ ; TMREM1
        BRA      GAI25
GAI24:  CLR.L    (A6)+
GAI25:  DBRA     D2,GAI24
```

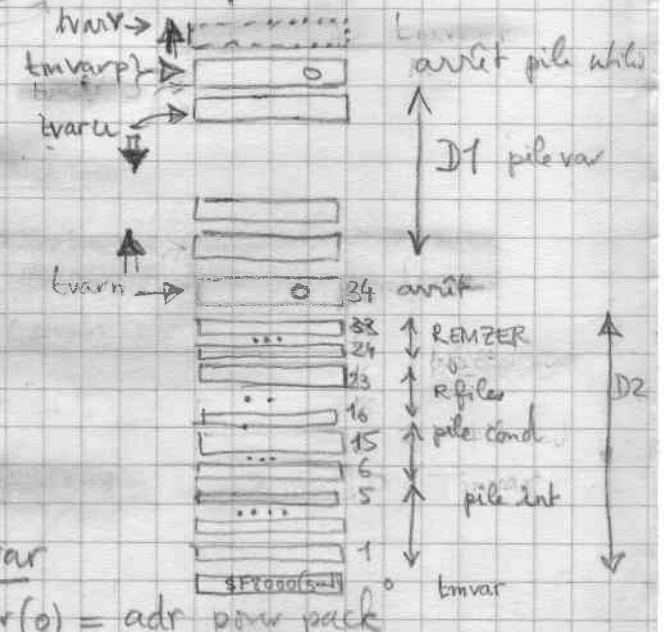


③

```
MOVE.L  A6,(A0)+        tmexc
MOVE.L  A6,(A0)+        tmexcp
CLR     (A6)+
ADD
MOVE.L  (A1)+,D0 A6     pile d'executions
ADD.L   D0,A6
MOVE.L  A6,(A0)+        tmcond
MOVE.L  A6,(A0)+        tmcondp
MOVE    (A1)+,D0        nb de condition - 1
PA25: CLR   (A6)+
      CLR.L (A6)+
      DBRA  D0,PA25
      MOVE  #-1,(A6)+   } arrêt d'empilement des conditions
      CLR.L (A6)+       ] fin de liste de table d'index
      MOVE.L A6,(A0)+   tmnom
      ADD.L TAMPNOM,A6  tampon noms
      MOVE.L A6,(A0)+   tmvarp
      CLR.L (A6)+       arrêt pile utilisateur

      MOVE  PILEVAR,D0  } → longueur de la pile - 1
      MOVE  D0,D1
      MOVEQ #PILEINT,D2
      MOVE  D2,TVARDC   } dernière condition utilisée
      ADD   PILECOND,D2
      CLR   TCMPX
      MOVE  D2,RFILES0  dernier R_files utilisé
      ADD   RFILES,D2
      MOVE  D2,-(SP)
      ADD   D2,D0
PA27: CLR.L (A6)+
      DBRA  D0,PA27
      MOVE.L A6,(A0)+   tmvar
      MOVE.L A7,(A6)+   var(0) = adr pour pack

      MOVE  D2,REMZER   1er n° table remember - 1 ⊗
      ADD   PILEREM,D2
```



n° de variable libre

```
MOVE.L #10, TBASET
MOVE. #$400A , (A0)+          tbase
) CLR PRIOR                   prior
ADDQ #1,D2

CLR.L (A0)+ TFRMT-2           ] tfrmt
      #34
MOVE #$22, (A0)+              } correspond à
MOVE #$28, (A0)+              } precision 10
MOVE #6 , (A0)+
                              MOVE #34, A2
                              MOVE.L #$22, TFRMTF-2
                                     A2
et déplacer tprec ... (6 octet)

MOVE D2, (A0)+                tvarndo
MOVE D2, (A0)+                tvarnd
MOVE D2, (A0)+                tvarn
ADD D1, D2    <- MOVE D2,TVARUM   valeur max ⊗
MOVE D2, (A0)+                tvaru
MOVE #1, (A0)+ <- MOVE #$7FFF,(A0)+   tvarl = 1   tvarls = $8000
ADDQ #2,D2
MOVE D2, (A0)+                tvarv
```

```
         TLIMIT
(MOVE.L (A0)+, (A0)+) -> inits      met tpile = tlimit

move.l $4BA.W ,ytime+6              timer

( st newlb      { met 0 labels utilisables
  bsr plmod1 et { 1  "   in  "        (source modifiée)
                                      (ou labels non mis)

( clr tdataf               } pointeur data (pour read)  -> inita
  move.l tdbas, tdatap )

inits <- [ clr.l tfonr      error ∅
           move #$14, GVALU ]
```

```
CLR.L  (A0)+        teppi
CLR.L  (A0)+        teppl

~~MOVE.L  A6,(A0)+~~   ~~tvar~~

CLR  (A0)+          tilde   {1 / 0 met ~

CLR.L  ORIGINX-2
CLR.L  ORIGINY-2
```

```
MOVE.L   TMVAR,A2
MOVE.L   #VARCMP,-(A2)          #1        i²+1  (calculs complexes)

                  {MOVE.L D2,TFRMTM       multiple des exposants
                  {CLR.L   TFRMTL         longueur des nombres AMM

MOVEQ  #1,D2
⊗ { MOVE (SP)+,D1         pileint + pilecond + Rfiler + remizer - 1 - 1
  { SUBQ #2,D1                                      v(1) déjà mis
  {                                                 + pileint - 1
  { MOVE   PILECOND,D1   ADDQ #pileint-2,D1    met PILECOND variables
                                 pour les conditions et pour π (n°2)
PA29 :MOVE.L  A6,-(A2)                                v(5) = ∅ [pour fsw et fsc]
                                                      v(n) = ∅
      ADDQ  #1,D2                           { CMP #5,D2
      MOVE  D2,(A6)+  <- PA292:MOVEL#4 ,(A6)+  BNE BCS PA292  ⊗  (et pileint=5)
                         MOVE.L #$14000,(A6)+   CLR.L (A6)+
                                            { BRA  PA294
PA294: BSR   VERA6
       DBRA  D1,PA29
       Move.L A6,(A0)+  ; tvala6  {CLR TPRECL   calcul de log2

       MOVEQ  #10,D3    } {precision 10
       BSR    MC534     } {format -10)
       CLR.L  TFRMTX-2    } format 0
       CLR    TPREC3
       BSR    XREPIC       } recalcule π et π/2
       MOVEL  A6,TVALA6

       MOVE.L #$4BA.W,YTIME+6     timer    ] ---> plus bas
       ST     TNEWLB       labels inutilisables
       JSR    PLMOD1
       BSR    XCLOSEA              ferme tous les fichiers
```

③ initialise TYPCAR   Explication ③5

```
        LEA     TYPCAR-2,A2
        CLR     (A2)+           -> met   MAJMIN=0  (variable d'état (m)distingue)
        CLR.B   (A2)+                    (min ≠ maj)
        MOVE    #254,D0
PA30:   MOVE.B  #$20,(A2)+
        DBRA    D0,PA30
        LEA     TYPCAR+$30,A2   <-- CLR.B  CARCOM(A2)   ";"  -$30
        MOVE    #9,D0
PA31:   MOVE.B  #$A6,(A2)+
        DBRA    D0,PA31         <-- { CLR.B  1(A2)  ";"
        LEA     TYPCAR+$41,A2
        MOVE    #25,D0          <-- MOVE  #$E6,D2
PA32:   MOVE.B  D2,32(A2)
        MOVE.B  D2,(A2)+
        DBRA    D0,PA32

        LEA     TYPCAR,A2         car α-num
        LEA     TYPCAR1,A0      <-- { MOVE.B  #$AE,$24(a2)   ($)   implicite chaîne
        BRA     PA311               MOVE.B  #$B6,$21(a2)   (!)   } index
PA310:  MOVE.B  #$A6,(A2,D0)        MOVE.B  #$B6,$25(a2)   (%)
PA311:  MOVEQ   #0,D0
        MOVE.B  (A0)+,D0
        BNE     PA310
        BRA     PA313
PA312:  MOVE.B  #6,(A2,D0)
PA313:  MOVE.B  (A0)+,D0
        BNE     PA312


        BSR     AESVDIPB   ①146.1
        BSR     VDIINIT              initialise vdi
        JMP     XCLRTIM    vide le temps
                                         $7F
traités comme α-num:   TYPCAR1: D.B  "#.?_",0 "âêîô éèç",0
traités comme lettres accentuées
```