③ Décode primary en A5

Met v(TVARN^e) = primary

TVARN^s ± 1

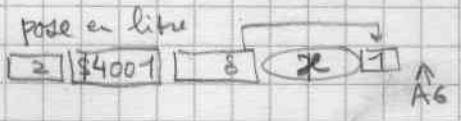
A5^s = avancé

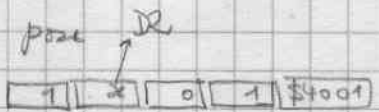
```
WPRIM:BSR    DECCRPG        (
x      BNE    LB121
       BSR    WEXPR
       BSR    DECCRPD        )
LB120: BNE    ERREX
       RTS

LB121: BSR    DECCRPH        [
       BNE    LB12
       BSR    WEXPR
       BSR    DECCRPE        ]
       BRA    LB120

~~LB12: MOVE.L  A5,-(SP)~~
LB12:  BSR    DECAN          nom α-numérique ?
       BNE    LB14           -> oui
~~MOVE.L  (SP)+,A5~~         type numérique

       ADDQ   #6,A6
       MOVE.L A6,-(SP)
       MOVE   PRIOR,(A6)+
       BNE    LB13
       CLR.L  (A6)+
LB13:  BSR    DECP           décode nombre et le pose en libre = {A2}^s
       BEQ    ERREX
LB130: MOVE.L (SP)+,A2       adresse
                             MOVE.L (A6),D0    fin
       MOVE.L D0,-(A2)       SUB.L  A2,D0     longueur
       MOVE   TVARN,D0       LB131: ADDQ #1,TVARN
       MOVE.L TMVAR,A0
       MOVE   D0,-(A2)
       EXT.L  D0
       ASL.L  #2,D0
       SUB.L  D0,A0
```

flottant ?

```
   tst.b    decpf
   beq      lb130          -> non (exact)
   MOVE.L   A2,A0
   MOVE.L   A5,-(SP)
x  BSR      XFLO           met <A2> = flottant
   MOVE.L   (SP)+,A5
   MOVE.L   (SP),A0        ({A0})
   MOVE     #-1,(A0)+
   BSR      XLB76
```

```
        TST.L   (A0)
        BNE     ERRPL          → erreur pile pleine
        MOVE.L  A2,(A0)

        RTS

LB14:~~ADDQ #4,SP~~            cas nom
LB14 :TST   D0
        BPL     LB19           → nom connu
                               nom inconnu : crée un littéral
        MOVE    #$10,D0        typ = literal
        BSR     WNWS           inscrit le littéral [na suivi de ( )]
                               x = D2^s (numéro)

LB16 :ADDQ  #6,A6                  crée V(TVARN) = x (de numéro D2)
        MOVE.L  A6,-(SP)
        CLR     (A6)+
        TST     PRIOR
x       BEQ     LB161
        SUBQ    #2,A6
        BSR     LB160
        BRA     LB130
LB161:  BSR     LB17
        BRA     LB130

LB160 : MOVE.L  #$24001,(A6)+
        MOVE.L  #12,(A6)+
        BSR     LB17
        MOVE    #1,(A6)+
        RTS

LB17 :  MOVE    #1,(A6)+
        MOVE    D2,(A6)+
        CLR     (A6)+
        MOVE.L  #$14001,(A6)+
        RTS
```





③ cas nom connu

```
LB19: MOVE.B  (A3)+, D0        type du nom
      MEVEN   A3
      CMP.B   #$10, D0
      BCC     LB30
```

cas fonctions internes

```
LB190: LEA    Y, A1
       CMP.B  #13, D0
       BCC    ERRRS
       ADD
       MOVE   (A3)+, D1 A1
       ADD    D1, A1
       TST.B  D0
       BNE    LB20
       JMP    (A1)            type sans argument
LB20:  CMP.B  #"(",(A5)+
       BNE    ERRAR
       MOVEM.L A1/A3, -(SP)
       CMP.B  #2, D0
       BGT    LB21
       MOVE   SR, -(SP)       1 et 2
       BSR    WEXPRE
       BRA    LB27
LB21:  CMP.B  #4, D0
       BGT    LB22
       MOVE   SR, -(SP)
       BSR    WPOLY
       BRA    LB27
LB22:  CMP.B  #6, D0
       BGT    LB23
       MOVE   SR, -(SP)
       BSR    WREEL
       BRA    LB27
```

③

```
LB23: CMP.B   #8,D0          flottant réel ou complexe
      BGT     LB24
      MOVE    SR,-(SP)
      BSR     ~~WMAT~~ WFLOAT
      BRA     LB27
LB24: CMP.B   #10,D0
      BGT     LB28
      MOVE    SR,-(SP)
      BSR     WCHAS
LB27: MOVE    (SP)+,SR
      BNE     LB28
      BSR     DECCRV     (5)
      BNE     ERRAR
LB28: MOVEM.L (SP)+,A1/A3
      JSR     (A1)
      BSR     DECCRPD    (7)
      BNE     ERRAR
      RTS

LB30: CMP.B   #$20,D0
      BCC     LB300
      BSR     WNUMVL       cas littéral
      BRA     LB16         → met D2 = numéro du littéral
```

×

③

```
LB300: CMP.B   #$30,D0
       BCC     LB32
```

Cas des index

```
       BSR     WNUMI          met {D2 A3 repèrent l'index
       TST     D2             met D0.L = valeur de l'index
x      BNE     LB301
       cas .B
       MOVE.B  (A3),D0        cas.B
       EXT     D0
x      BRA     LB303
LB301: CMP     #3,D2
       BNE     LB302
       MOVE.L  (A3),D0        cas.L
x      BRA     LB305
LB302: BCC     LB304
       MOVE    (A3),D0        cas.W
LB303: EXT.L   D0
       BRA     LB305
LB304: MOVEQ   #0,D0
       MOVE.B  (A3),D0
       ASL     D2,D0
       EXT     D0
       EXT.L   D0
       ASR     #8,D2          ⊗ {4 1/2, 6 1/4, 7 1/8
       ASR.L   D2,D0
LB305: ADDQ    #6,A6          crée V(TVARN) = D0.L
       MOVE.L  A6,-(SP)
       BSR     WSD3L
       BRA     LB130
```