

return [label]

```

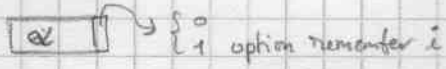
YRET: BSR CLRLOC @186   vide les variables locales A0 = (TMPROCA)
      MOVE -(A0), D0     k indicateur du sous-progr
      BEQ  ERRGRT        return sans appel
      MOVE -(A0), D1     n nb d'arguments
      MOVE.L -(A0), A2   apres l'appel

      ADD  D1, D1
      ADD  D1, D1
      SUB  D1, A0
      CMP  #1, D0
      BEQ  MAS0           -> procedure
      CMP  #5, D0
      BEQ  MAS0           -> SP 68000
  
```

fonction (3.4.4)

```

      MOVE.L -(A0), D1
      BTST #0, D1
      BEQ  GAI30
  
```

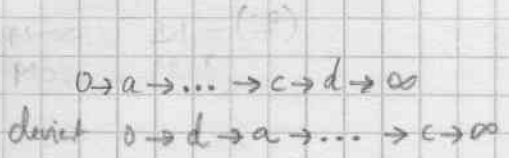


remember i

d1 = alpha

```

      MOVEM.L D0/A0/A2, -(SP)
  
```



```

      MOVEM.L (TMPROCA), A0/A1
  
```

```

      MOVE (A0), d2      d
  
```

```

      MOVE.L (A0), d0     a
  
```

```

      MOVE (a0, d2.w), d3  c
  
```

```

      CLR 2(a0, d3.w)     c | x | 0
  
```

```

      MOVE.L IO, (a0, d2.w) d | 0 | a
  
```

```

      MOVE d3, (a0, d0.w) a | d | x
  
```

```

      MOVE.L d2, (a0)     } | 0 | c | d
  
```

```

      MOVE d3, (a0)      c
  
```

```

      MOVE.L d1, -(a1, d2.w) d | alpha
  
```

```

      BSR REMVAR         } no de la variable
  
```

```

      MOVE D1, -(SP)
  
```

```

      MOVE GVALU+2, d2   } pousser value
      BSR LB321
  
```

```

      MOVE (SP)+, d0
  
```

```

      BSR WASGNR
  
```

```

      MOVEM.L (SP)+, d0/a0/a2
  
```

```

GAI30: MOVE.L -(A0), GVALU
        MOVE.L TPILE, SP
        MOVE.L -(A0), TPILE

```

```

MAS0: MOVE.L -(A0), D1      ancien TMFOR

```

```

        LEA TMFOR, A1

```

```

        CMP.L (A1), D1

```

```

        BCS ERR TLP

```

```

        MOVE.L D1, (A1)

```

```

        MOVE -(A0), TVARLS

```

```

        MOVE.L -(A0), D1

```

```

        BRA MAS03

```

```

MAS02: MOVE.M.L Do/D1/A0-A2, -(SP)

```

```

MAS02: BSR POPN

```

```

        MOVE.M.L (SP)+, Do/D1/A0-A2

```

```

MAS03: MOVE.M.L Do/D1/A0-A2, -(SP)

```

```

MAS03: CMP TVARN, D1

```

```

        BNE MAS02

```

```

        MOVE.L D1, TVARND

```

```

        CMP #5, D0

```

```

        BEQ MAS10          68000

```

```

        CMP #1, D0

```

```

        BEQ MAS1

```

```

        MOVE.L A2, A5

```

```

        BSR BCKTMP

```

```

        MOVE.L A5, TDEBAS

```

```

        CMP.L YRETC, A0

```

```

        BCC MAS05

```

```

        MOVE.L SP, YRETC

```

```

MAS05: RTS

```

```

MAS1: BSR BCKTMP

```

```

        BSR DECTMN

```

```

        BNE YGOTO

```

```

        MOVE.L A2, A5

```

```

        BSR DEPILE2

```

```

BRA LB93

```

```

MAS10: BSR BCKTMP

```

```

        BSR LB920

```

```

        BSR MA512

```

```

        JMP (A2)

```

```

DEPILE2: BSR LB920

```

```

        BRA DEPILE1

```

⊗

le S-P a termine de boucles

rectifie la pile

```

MOVE.M.L Do/D1/A0-A2, -(SP)
ADDQ #1, D1
BRA MAS03

```

⊗ (D1) contient U(5) <sup>mémor</sup>

```

MAS02: MOVE D1, -(SP)
        BSR POPN
        MOVE (SP)+, D1
MAS03: CMP TVARN, D1
        BNE MAS02

```

```

MOVEQ #5, D0
BSR WASGN2
MOVE.M.L (SP)+, Do/D1/A0-A2

```

remet U(5) pour boucles for/fore

```

        MOVE.L A0/A1, -(SP)
        → proc
        BSR DEPILE2
        MOVE.L (SP), A0/A1

```

⊗ 16

dépile

instruction en cours

pour debug ⊗ ok si 2 appels de la m fonction dans une expression

→ return label

BSR DEPILE2 ⊗ dépile

return d'appel dans code 68000

) ⊗