

```

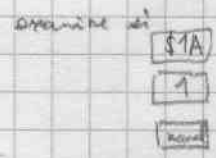
WINSTR : CMP.B #1, (AS)
           BNE PB79
           BSR WEXCEND
           MOVE.L DO, AS
           ↓
           PB79
    
```

dépile instruct. de la pile exec



```

WINSTR : MOVE.B (AS)+, DO
           CMP.B #1, DO
           BHI PB79
    
```



```

PB78 : BEQ WINSTR YEND
           CMP.B #1, DO ← BCS WINSTR (cas 0)
           BNE ERRFAT
           BSR WEXCEND
           MOVE.L DO, AS
    
```



```

WINSTR : MOVE.B (AS)+, DO
           PB795 : CMP.B #1, DO
           BLS PB78
    
```

```

PB794 : MOVE.L AS, AD
           MOVE #10, D1
           ↓
           PB80
    
```

③ traduit instruction en AS et l'exécute  
ERRARV : MOVE #, D0 → rer err valeur de l'argument

ERRPXC : MOVE #, D0 → rer err pile epee trop petite  
ERRARN : # → rer er ~~#~~ interdit ds main

PB78 : BRA WINSTR1

PB79 : MOVE.L A5, A0

MOVE #10, D1

PB80 : MOVE.B (A0)+, D0  
BEQ.S PB78 WINSTR1 → non  
CMP.B D0, D1  
BNE PB80

traduc

LEA TMEXCP, A1

MOVE.L (A1)+, A0

MOVE.L (A1), A2

SUB #16, A2

} il faut A0 < A2  
Imarge de fin

A0 A1 A2, A  
D1 A

BRA PB82

PB81 : CMP.L A2, A0

BCC ERRPXC

MOVE.B D0, (A0)+

err pile epee trop petite

PB82 : MOVE.B (A5)+, D0

BEQ PB88 → fin

CMP.B D0, D1

BNE PB81

CMP.B (A5)+, D1

BEQ PB81

SUBQ #1, A5

@@  
est transféré en @

repite WADR

MOVEM.L D1/A0-A2, -(SP)

BSR WPRIM

← BSR DECCRAS saute 95

BSR XIADR

BNE ERRARV

BSR POPN

@  
primary mis dans la pile  
teste si top pile est adresse

MOVE.E T<sup>A</sup>MPROC, A3

tmproc

MOVE -(A3), D4

{ 1 proc  
2,3 touch

BEQ errarn

MOVE -(A3), D0

} err  
n

EXT.L D0

```

(3) MOVEM.L (SP)+, D1/A0/A1/A2
TST.L D3
BNE PB84

```

ce no, écrit le nombre d'arguments D0.L en A0

MOVE.B # "g", (A0)+ en décimal

x BSR PINTA avance A0, conserve le reste

```

MOVE.B # " ", D0
BRA PB81

```

```

PB84: BMI ERRARV
PB84 SUB.L D3, D0
BCS ERRARV

```

```

BSR PB840 @
CMP #1, D4
BNE PB85

```

```

PB840: ASL.L #2, D0
SUB.L D0, A3
ADD.L -(A3), D0
MOVE.L -(A3), A3
SUB.L A3, D0

```

$D0 = \begin{cases} -1 & \text{arg 1, 2, \dots, n-1} \\ 0 & \text{arg n} \end{cases}$  broc  
 $D0 = 1$  : arg 1, 2, \dots, n fonct

début de l'argument  
longueur de l'argument

```

lea (A0, D0.L), A4
cmp.l A2, A4
bcc ERR PXC

```

```

PB86: MOVE.B (A3)+, (A0)+
DBRA D0, PB86
SUBQ #1, A0
BRA PB82

```

PB88: BSR PB880  
BRA WINSTR1
] mis devant WINSTR1

```

(5) PB885: MOVE.B # " ", (A0)
MOVE.L A0, D0
BSET #0, D0
MOVE.L D0, A0
CLR.B (A0)+
MOVE #100, (A0)+
MOVE.L A5, (A0)+
MOVE.L -(A1), A5
MOVE.L A5, (A0)+
MOVE.L A0, (A1)

```

tmaxcp initial  
début de l'instruction de specs  
nouveau tmaxcp

```

PB881: BRA WINSTR1 RTS

```

si l'argument k est	@R(etc) devant
forme(a)	forme(a, etc)
forme	forme(etc)
forme()	forme(, etc)

```

PB86: MOVE.B (A3)+, (A0)+
      DBRA D0, PB86
      SUBQ #2, A0
      CMP.B #" ", (A0)+
      BNE PB82      →
      CMP.B #"(", (A5)
      BNE PB82      →
      ADDQ #1, A5
      MOVE.B #" ", -1(A0)
      BRA PB82
  
```

l'argument se termine par 0 ?

l'instruction continue par 0

↓ oui changer **2** en **3**