

③ Décode chainess  
et pousse ds la pile

||        ||  
a5  
aL  
aZ  
cfonctn } noms  
chain }  
expr

( "" pour nul )  
chr (32)  
chr (13)  
chr (0)

```
WCHAS1: BSR DECCRG
        BNE LC13
        ADDQ #6, A6
        MOVE.L A6, A2
```

Ⓜ

ici d1=34 Ⓜ a été mis par DECCRG

ⓧ

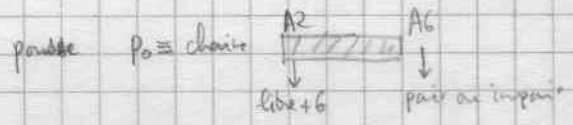
```
LEA -1000(SP), A0
BRA LC11
LC10:  CMP.L A0, A6
        BCC XERRM      erreur minime
        MOVE.B D0, (A6)+
LC11:  MOVE.B (A5)+, D0
        BEQ ERRIS
        CMP.B D1, D0
        BNE LC10
        CMP.B (A5)+, D1
        BEQ LC10
        SUBQ #1, A5
```

erreur minime

Ⓜ → Ⓜ

```
LC10:  MOVE.B (A5)+, D0
        BEQ ERRIS
        CMP.B #32, D0
        BEQ LC12
LC11:  MOVE.B D0, (A6)+
        BRA LC10
```

```
Ⓟ LC12: MOVE.L A6, D1
        SUB.L A2, D1
        MOVE.L D1, -(A2)
        BRA LB991
```



```
LC13:  MOVE #32, D2
        BSR DECCRAS
        BEQ CPUSHN1
        MOVE #13, D2
        BSR DECCRAL
        BEQ CPUSHN1
        CLR D2
        BSR DECCRAZ
        BEQ CPUSHN1
```

Ⓟ

Ⓟ

Ⓟ

```

MOVE.L A5, A4
BSR DECAN
BNE LC18

```

nom de num?  
qui

```

LC16: MOVE.L A4, A5
BSR WEXPR

```

```

YSTR: BSR LB95A XIFLOA
BMI err fut ; *A

```

A2 pointe variable sur la pile

```

MOVE.L A2, A0
ADDQ #5, A6
ST (A6)+
MOVE.L A5, -(SP)

```

```

BSR XPRTAF
MOVE.L (SP)+, A5
MOVE.L A0, A2
MOVE TVARN, D0
BRA PB430

```

pose en libre la chaîne  
S=A0

```

LC17: TST.B -(A6)
BEQ LC17
ADDQ #1, A6

```

⊗ (après PB43 de XHCHA)

LC18:

```

LC18: TST D0
BMI LC16
MOVE.B (A3)+, D0
MEVEN A3
MOVE D0, D1
AND #F0, D1
CMP #40, D1
BEQ LB320
CMP.B #54, D0
BEQ LC19
CMP.B #60, D1
BNE LC16 LC182
SUB D1, D0
BRA LB190
LC182: SUB.B #78, D0
BNE LC16
BRA LB190

```

→ na inconn  
type des num

→ nom de chaîne

→ fonction externe chaîne

fonction interne string?

→ suppose que c'est expression

→ fonction interne

→ Clavier ou commande

X

⊗

⊗

③ fonction externe chaîne

```

LC19: MOVEQ #0, D3      } pause chaîne vide
      BSR    CPUSHNO        }
      MOVE  #3, -(SP)      } type chaîne
      BRA   LB36

```