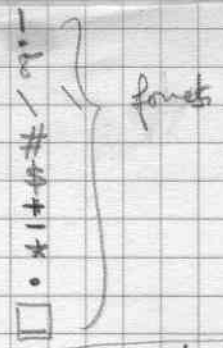
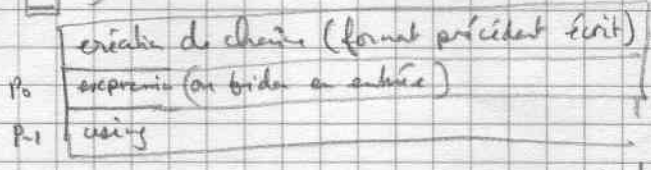


installe using : écrit & avant
 écrit



entrée : A1 E format using (en p₀-1)
 A6 libre



- a) Écrit en A6 tout ce qui est avant un format, puis crée p₀ avec ^{suivant A2} et ôte p₀
- b) Si numérique modifie FORMATX etc...

Sortie ^{nouvelle} points de A1 marquée sur p₀

début
 d0/d1/d2/d3 a0/a1

```

(SP) GAM35: MOVE.B #0, (A6)+
    USING CH: MOVE.B (A1)+, d0
    BEQ GAM40
    
```

```

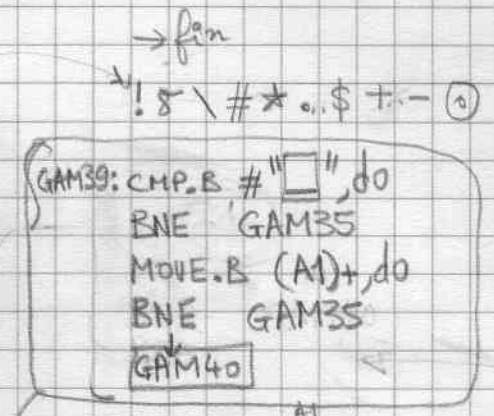
(GAM36:) LEA GAM36, A0
    
```

```

GAM38: MOVE.B (A0)+, d6
    BEQ GAM39
    CMP.B d1, d0
    BNE GAM38
    
```

```

SUBQ #1, A1
GAM39: MOVE.L A1, -(SP)
    MOVE.L A1, d2
    BSR XILOC
    SUB.L A1, d2
    ..
    MOVE.L (SP)+, A0
    BRA GAM42
    
```



(contient le format avec l'offset au début)

```

GAM40: BSR XILOC
    MOVEQ #6, d2
    MOVE.L A1, A0
    ADDQ #6, A0
    
```

cas format terminé: remet au début

```

GAM42: MOVE.L d2, (A1)
    BSR USING1 543 ← CUR.B (a6)+ ⊗
    BSR LC12
    BRA POPPR
    
```

```
GAB98: TST 4(SP)
      BEQ  GAB985
```

using?

→ non

```
GAB99: BSR  WTSTCH
      BEQ  GAB983
```

vaie chaîne ?

→ oui

```
BSR  WCHASZ
```

⊙

```
BSR  USING2
```

```
BRA  GAB986
```

```
GAB983: BSR  WCHAS
        BSR  USING3
        BRA  GAB986
```

```
GAB985: BSR  WCHAS
```

```
GAB986: MOVE.L (SP), D6
```

```
TST 4(SP)
```

si "LBS:11

```
BPL  GAD10
```

```
MOVE.B D6, REGD7+3
```

```
BSR  YCHAN1
```

```
GAD10: BSR  HIDECH1
```

⊗

GADR: BSR PACK → (en cas de "MEM.")

MPROTZ : BSR LB95A

MOVE.L A0, A3

TST 4(SP)

BLE GAD16

MOVE.L LEA SP, A0

MOVEM.L D5/A3, -(SP)

BRA GAD15

GAD14: MOVE.B (A3)+, D0

BSR INCPOS

GAD15: SUBQ.L #1, D3

BPL GAD14

MOVEM.L (SP)+, D5/A3

BRA GAD30

GAD16: BNE GAD30

BTST #3, 3(SP)

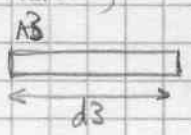
BEQ GAD30

BRA GAD22

GAD19: BSR GAC39
MOVE D1, 10(SP)

BSR X9

BSR HIDECM



si autre que VBS: LBS: modifie pos

nb de lignes. non -1

cas "VBS:" et /A/

sortir d5+1 lignes à partir de A3



5
repete
M131

```
GAD22: TST.L D3
GAD23: BLE GAD32 → fin
```

x

```
TST.B TESC.V
BNE GAD25
MOVE.B (A3), D0
BEQ GAD24
CMP.B #13, D0
BEQ GAD24
LEA TCURSC, A1
MOVE (A1)+, D2
CMP (A1), D2
BNE GAD25
```

```
GAD24: SUBQ #1, 10(SP)
      BMI GAD27
```

{ GAD25: TST 10(SP)
 → message

```
GAD28: MOVE.B (A3)+, D0
      BSR X1RE
      SUBQ.L #1, D3
```

sortie VBS:

x

```
BRA GAD23
```

```
GAD27: BSR XARRET
      BNE GAD19
      BSR POPN
```

repete gad 84 →

```
BSR GAC26
ADD #14, SP
MOVEM (SP)+, D0/D1/D3/D4
```

de using

```
BSR GAB85
BRA YEND
```

remet format/bases

⊗

GAD30: MOVE.L 6(SP), A0 carnal cas autre que VBS: et /A/
 BSR PTCAN sortie

GAD32: BSR POPN fin de sortie de chaîne

BSR DECTMN fin ?
 BEQ GAB83

BSR DECCRV (g) ou (j)
 BNE ERRIS8 → erreur
 SUBQ #1, A5

GAD33: BSR DECCRV (g)
 BNE GAD334 → non

TST 20(SP)
 BNE GAD336 → write

TST 12(SP)
 BNE GAD332 → using

BSR GAC13 do = pos tabulation
 AND #\$F, D0
 MOVEQ #16, D3

SUB D0, D3
 BSR GAC18 sort d3 espaces

GAD332: BSR DECTMN suite de fin ?
 BEQ GAB84 → ne va pas à la ligne
 BRA GAD33

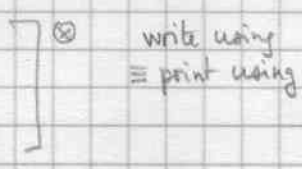
GAD334: BSR DECCRV (j) ?
 BNE GAB82 → non
 TST 20(SP)

BEQ GAD332 → print

GAD336: TST 12(SP) ↓ write

BNE GAD332 → using

write ~~GAD336~~: BSR GAC181 écrit (g)
 BRA GAD332



SP Z pile mel do = $\left\{ \begin{array}{l} \text{cursx} \\ \text{lstpos} \end{array} \right.$ si VBS:
si LBS:
sinon

SP GAC13: TST 8(SP)

BEQ GAC15
BMI ~~GAC14~~ XLPOS
MOVE 4(SP), DO
RTS

GAC14: MOVE LSTPOS, DO
RTS

GAC15: MOVE TCURSX, DO
RTS

XLPOS ~~MOVE~~ #0, DO
~~GAC14~~ MOVE.L #0, DO ; L
LSTPOS = * - 2
DIVU IMPX, DO
SWAP DO
RTS \leftarrow MOVE DO, LSTPOS

à la ligne

SP GAC16: MOVE.L 10(SP), A0
BSR PTCAN13
CLR 4(SP)
GAC17: SUBQ #1, 14(SP)
RTS

canal
13, 10
pos
nb de lignes: -1

ant d3 espaces
GAC18: MOVEQ #32, DO

SP GAC18: MOVE.L 10(SP), A0
ADD D3, 4(SP)
BRA GAC20

canal
pos
LEA TCURSX, A1
MOVE (A1), D1

~~GAC19: MOVEQ #32, DO~~
GAC19: BSR PTCANB
GAC20: DBRA D3, GAC19
RTS

CMF (A1), D1
BGT GAC17 \rightarrow changement de ligne

ste po si USING = 1 et met USING / 0

SP GAC26: TAS 16(SP)
BNE POPN
RTS

ant caractère do (" ")

GAC181: MOVEQ #",", DO
GAC182: MOVEQ #1, D3
BRA GAC180

⊗

examine si (terminal)

oui EQ
non NE

✓

```

(SP) GAC22: BSR DECCRPV
      BNE GAC23
      MOVE.L AS, -(SP)
      BSR DECTMN
      BEQ GAC24
      MOVE.L (SP)+, AS
      SUBQ #1, AS
GAC23: RTS
GAC24: ADDQ #4, SP
      RTS

```

[entrée A0 (using)]

Avance A0 jusqu'au 1ere # ou * ou @ ou GA

```

(SP) GAC28: MOVE.B (A0)+, do
      BEQ GAC30 → rtb
      CMP.B #"-", do
      BNE GAC29
      MOVE.B (A0)+, do
      BEQ GAC30 → rtb
      BRA GAC28
GAC29: BSR GAC52 # * ou .
      BEQ GAC30
      CMP.B #"-", do
      BNE GAC28 → nm
GAC30: SUBQ #1, A0
      RTS

```

↓ sans
↓ saute suivant

avance ad jusqu'au \odot ou fin de format numérique
sauts # * L 9 et s'arrête sinon

(SP) GAC31: MOVEQ #".", d1

GAC32: MOVE.B (A0)+, d0

BEQ GAC30

BSR GAC53 * ou #

BEQ GAC32 → oui

CMPI.B d0, d1

BEQ GAC30 → \odot trouvé

BSR GAC56 9 ou W

BEQ GAC32 → oui

MOVEQ #0, d0

BRA GAC30 → pas de 0 dans le format

Met pour N/ le nb de lignes max - 1 → d1^s

(SP) GAC34: LEA TCURSB, A1

#MOVE (A1), D1

SUBQ #1, D1

SUB -(A1), D1

RTS

~~avance ad jusqu'au \odot~~ (d)
GASTO:

5

en A0 Compte les # (saute L et g) jusqu'à autre chose que L ou g
d3 = valeur ^{et *} et n

(SP) GAC40: MOVEQ #0, D3 ✓
BRA GAC44

GAC42: ADDQ.L #1, D3

(SP) GAC44: MOVE.B (A0)+, D0
BEQ GAC46
BSR GAC53 * ou #

BEQ GAC42
BSR GAC55 n | g ou L

BEQ GAC44

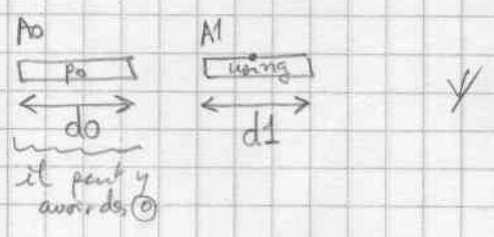
GAC46: SUBQ #1, A0
RTS

5
P0 à traduire
P1 format

} à traduire

80

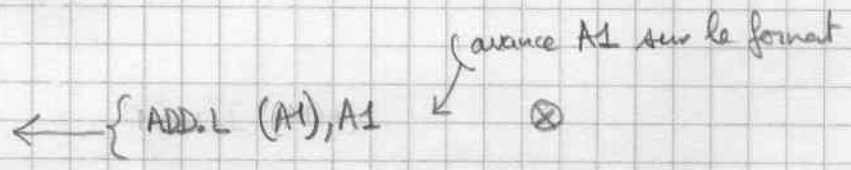
GAC50:BSR X1LOC



```

ADDQ #6, A6
MOVE.L A6, A2
ADD.L D0, A6
ADD.L D1, A6
BSR VERAG
MOVE.L A2, A6

```



RTS

lets do.B = #
.*

```

(SP) GACS1: CMP.B #"~", do
      BEQ   GACS4
  
```

{ CMP.B #"^", do } @
 { beq gacs4 }
 { CMP.B #"_", do }
 { BEQ GACS4 }

```

(SP) GACS2: CMP.B #".", do
      BEQ   GACS4
  
```

. # *

```

(SP) GACS3: CMP.B #"#", do
      BEQ   GACS4
      CMP.B #"*", do
GACS4: RTS
  
```

*

```

(SP) GACS5: CMP.B #"~", do
      BEQ   GACS4
  
```

(~) (~) (~)

```

(SP) GACS6: CMP.B #"g", do
      BEQ   GACS4
      CMP.B #"u", do
RTS
  
```

int \$/signe seival d5

```

(SP) GAC58: CLR D1
      TST DS
x    BPL GAC59
      int $
      MOVEQ #1, D0
      } MOVE.L DS, D0
      SWAP D0
      }
      GAC59: BSR GAC61
  
```

```

GAC59: BTST #8, DS
      BNE GAC60
      signe ⊖
      } CMP.B #2, DS
      BEQ GAC54
  
```

```

(SP) GAC59: MOVEQ #"-", do
      BRA GAC61
  
```

```

x    GAC60: CMP.B #1, DS
      BNE GAC554
      MOVEQ #"+", do
  
```

```

(SP) GAC61: MOVE.B do, (a6)+
      SUBQ.L #1, d4
      CMP.L DS, D4
      BGE GAC554
  
```

```

(SP) GAC60: SUBQ.L #1, D3
      MOVE.B (a1), do
      BSR GAC51
      BNE GAC554
  
```



```

GAC62: ADDQ #1, A1
      MOVE.B (A1), do
      BSR GAC56
      BNE GAC554
  
```

```

GAC62: MOVE.B do, (A6)+
      BRA GAC62
      } TST D1
      BEQ GAC620
      MOVE D1, D0
  
```

GAC6: RTE