

décode bloc

ad A	tout	D1	D0	D2
ad B	ancien	1 (bloc trouvé demi)	per ligne si débloc = 0 mis 1	fin du bloc = ligne + 1 limité à eof
n, v				
ligne, ligne ($\neq n, v$)				
ligne, +n				
ligne \otimes ($x \neq y$) ligne $\neq \emptyset$		2 (ent ligne)	1 ligne (pas une une)	ligne + 1 if ≥ 2
\emptyset		0	0	eof
erreurs		-6	erreur ordre ligne $<$ ligne ou overflow de n, v	

défini tout sauf D7

WDBLK: WDAVR D0

```

SUBQ.B #1, D0 CMP.B #90, D0
BNE HD10
MOVEQ #1, D0
MOVE.L TDLGN, D2
BRA G981
    
```

HD10: SUBQ.B #1, D0 CMP.B #90, D0

```

BNE HD11
LEA TPLC, A4
MOVE.L (A4), D0
MOVE.L 8(A4), D2
BRA G981
    
```

HD11: MOVE.L A6, A4 HD11: SUBQ.L #1, A6

```

BSR WDN
BNE HD19
CMP.B #"9", (A6)+
BNE HD19
    
```

ADDDQ.L #1, A6

```

MOVE.L D0, D4
MOVE D1, D5
BSR WDN
BNE HD19
CMP D5, D1
BGE HD19
    
```

décode n
non deb=A4
form n
non deb=A4
suivi de v
D4 = valeur de n
D5 = longueur de n
décode v
non deb=A4
non deb=A4

création de $n_1 10^{D1}$ sur TDEQ1

$n = \begin{matrix} n_2 & n_1 \end{matrix}$

```
MOVE.L A6, A3    sauvegarde de A6
MOVE.L A4, A1    debut de n
```

$n = n_1 10^{D1} + n_2$
 $n' = n_1 10^{D1} + v + \begin{cases} 0 & \text{si } v \geq n_2 \\ 10^{D1} & \text{si } v < n_2 \end{cases}$

```
LEA TDEQ1, A2
MOVE.L A2, A6
```

```
HDR: CMP D5, D1    chiffres de n1
     BEQ HD13
```

```
HD13: ASL #2, D1    D1 = D1 x 4
```

```
NEG D1
LEA XE104, A0    table des 10x
MOVE.L (A0, D1.W), D5    D5 = 10-l
```

```
HD14: MOVE.B #0, (A2)+    met le "0" X
```

```
ADDQ #4, D1
```

```
BNE HD14
```

```
CLR.B (A2)
```

```
MOVE.L D0, D6    D6 = v
```

```
BSR WDN
MOVE.L D4, D1    D0 = n1 10l
```

```
SUB.L D0, D1    D1 = n2
```

```
ADD.L D6, D0    D0 = n1 10l + v
```

```
CMP.L D1, D6    BCS HD18 → overflow
```

```
BGE HD15
```

```
ADD.L D5, D0    D0 = n1 10l + v + 10l si v < n2 → overflow
```

```
HD15: MOVE.L A3, A6    ici D0 = ligne 2
      MOVE.L D4, D2    D2 = ligne 1
```

```
HD18: TST.L D0 HD15: EXG D0, D2    teste le bloc
      BNE HD16    MOVE #1, D1
```

si 1^{ere} ligne = 0, met D0 = 1

```
ADDQ.L #1, D0
```

```
BCS HD18 HD16: SUBQ.L #1, D2
      MOVEQ #2, D3
      ADDX.L D3, D2
```

```
LEA TDLGN, A0
CMP.L (A0), D2
BLE HD17
```

si D2 > max+1 → D2 = max+1

```
HD17: MOVE.L (A0), D2
      CMP.L (A0), D0
      BLE HD17
      MOVE.L (A0), D0
```

HD170: CMP.L D0, D2

compare D0 et D2

BCS HD18

→ erreur d'ordre

~~BRA HD181~~ ~~CMP D1, D1~~
~~RTS~~

HD18: MOVEQ #-6, D1

overflow) ou erreur de n, v) ligne2 < ligne1

HD181: MOVE.L A4, A6

HD180: RTS

HD19: MOVE.L A4, A6

BSR WDLGN

decode ligne

BNE HD180

erreur

WDAVR D2

car suiv

CPM.B #"g", D2

suivi de virgule

BEQ HD21
TST D1 ← SUBQ.L #1, A6

BNE HD20

cas ∅ bloc = all
↓ (D1=0 déjà mis)

MOVE.L TDLGN, D2

CLR.L D0

RTS

HD20: MOVE.L D0, D2 ← MOVE #2, D1
BRA HD16

cas 1. seule ligne donnée

HD21: MOVE.L D0, D5

D5 = 1er ligne

WDAVR D2

car suiv

CPM.B #"+", D2

BEQ HD23

SUBQ.L #1, A6

cas ligne1, ligne2

BSR WDLGN

BNE HD181

→ erreur

TST D1 ← MOVE.L D5, D2 ligne2

BNE HD22

MOVE.L TDLGN, D0

HD22: BRA HD150

MD23: BSR WDN

BNE MD181

MOVE.L D5, D2

ADD.L D2, D0

SUBQ.L #1, D0

BRA MD22

cas ligne₁, +n