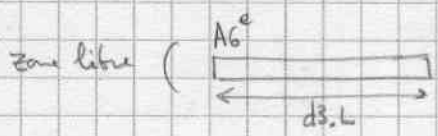


5

Entrée canal A0



(le type est supposé ok en lecture)

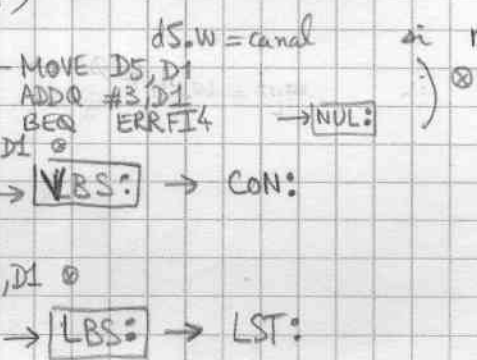


Sortie lit d3.L octets sur le canal A0 et les met en A6  
Si EOF -> erreur

conservé A0, avance A6 en  $A6 + D3^e$   
détruit A1-A2  
D0-D5

```
RDCAN1: ADD.L D3, A6
        BSR VERAG
        SUB.L D3, A6
RDCAN:  MOVEM.L D3/A0, -(SP)
```

```
RDCAN:  BSR XCANAL1
        MOVEQ #2, D0
        CMP # -1, D5
        BEQ GAS8
        MOVEQ #0, D0
        CMP # -2, D5
        BNE GAS9
        GAS8: MOVE D0, D5
        GAS9: BPL GAG1
```

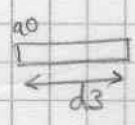


si mem: a1 pointe sur TMEM + (8-d5)

x

mem:

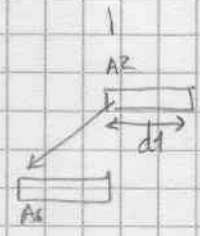
```
MOVE (A1)+, d4
BSR LB95C
ADD.L (A1), A0
SUB.L (A1), d3
BLE ERRFI8
MOVE.L (SP)+, D1
CMP.L D3, D1
BHI ERRFI8
MOVE.L A0, A2
```



end of file

end of file

avance le pointeur



```
GAG0: MOVE.L (SP)+, A0
RTS
```

```
GAS90: MOVE.B (A0)+, (A6)+
        SUBQ.L #1, d1
        BNE GAS90
```

```

GAG1: CMP #6, D5
      BCC GAG4
      [0-5] parts

```

```

GAG2: MOVE D5, -(SP)
      MOVE #2, -(SP)

```

```

GAG2: TRAP #13
      MOVE.B D0, (A6)+
      SUBQ.L #1, D3
      BNE GAG2
      ADDQ #4, SP
      BRA GAG0

```

← ADDQ #4, SP

⊗ ôte aussi d3

[disque handle = d5]

```

GAG4: MOVE.L A6, -(SP)
      MOVE.L D3, -(SP)
      MOVE D5, -(SP)
      MOVE #03F, -(SP)
      TRAP #1
      TST.L D0
      BMI ERRDOS
      CMP.L D0, D3
      BNE ERRFIS
      ADD #12, SP
      ADD.L (SP)+, A6
      BRA GAG0

```

end of file