

teste si Do.L = alt + chiffre
 prepare D1 = []
 et D0

NE: non
 EQ: oui

valleur = D2 (0-9 chiffre)
 \$FF esc
 \$23 H

D0 = W =



```

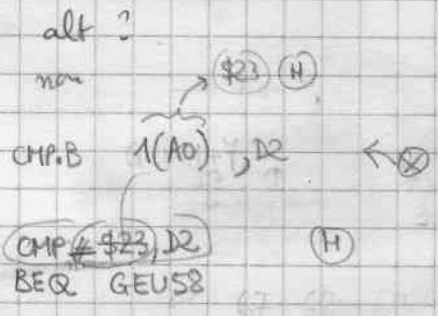
XKEYAL: MOVE.L D0, D1
        CLR D1
        ASR.L #8, D1
        OR D1, D0
        SWAP D1
        BCLR #1, D1
        BEQ GEU53
        BSET #0, D1
GEU53: BCLR #3, D1
        BEQ GEU54
        BSET #1, D1
  
```

```

* lea gen615, a0
  moveq #5, d5
gen56: cmp.b (a0)+, d2
       bne gen57
       moveq #15, d2
       sub d5, d2
       bra gen58
gen57: dbra d5, gen56
  
```

```

GEU54: AND #7, D1
        CMP #2, D1
        BNE GEU60
XKEYLC: MOVE D0, D2
        LSR #8, D2
        SUBQ #1, D2
        CMPB #10, D2
        BCS GEU58
        BEQ GEU59
  
```



x
 x
 x
 x
 x

```

SUB #10, D2
CMPB #10, D2
BCC GEU60
MOVE D0, D2
SUB #30, D2
GEU58: AND #FF, D2
        OR #4, SR
        RTS
GEU59: MOVEQ #0, D2
        RTS
  
```



eg vrai

```

GEU60: MOVEQ #-1, D2
        RTS
  
```

GEU60: D.B 1, \$78, \$79, \$7A, \$7B, \$7C, \$7D, \$7E, \$7F, 0

\$10	\$30	\$2E	\$20	\$12	\$21	\$22	\$23	\$17	\$24	\$25	\$26	\$27
A	B	C	D	E	F	G	H	I	J	K	L	M
\$31	\$18	\$19	\$1E	\$13	\$1F	\$14	\$16	\$2F	\$2C	\$2D	\$15	\$11
N	O	P	Q	R	S	T	U	V	W	X	Y	Z

la code chiffre
 et le code \$23 (H)
 Set aussi
 valeurs
 clavier pour
] clavier <-
 curso

```

GEU61: BLK.B 26
  
```

tranche de tranchi Fk ?

```

GEU63: MOVE D0, D2 ← MOVEQ #10, D3
      LSR #8, D2 ← MOVE D2, D4
      SUBB # $3B, D2
      CMP.B D3, D2
      BCS GEU64 → oui
      SUBB # $19, D2
      CMP.B D3, D2
      BCC GEU66 → non

```

```

GEU64: MULL D3, D1
      ADD D1, D2
      ADD # $200, D2

```

```

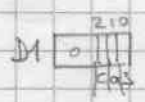
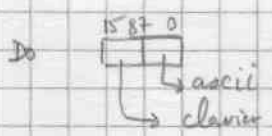
GEU65: MOVEQ #0, D0
      MOVE D2, D0
      BRA GEU46

```

oui :

$$D.O.L = \$200 + 10(4c + 2a + s) + R - 1$$

E[0, 80E]



```
GEU66: CMP #4, D1
```

```
BNE GEU69
```

Control + lettre ?

```
BSR XKEYLC
```

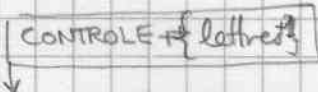
```
BNE GEU680
```

→ non

```
MOVE D2, D3
```

```
ADD #50, D3
```

⊗



```
MULU #100, D3
```



∈ [4, 25]

recopie GEU48

```
GEU67: BTST #2, (A3)
```

```
BEQ GEU68
```

→ control relaché

```
MOVE.B #6, T193+3
```

```
BSR ESCAPE
```

```
BSR RD
```

```
BEQ GEU67
```

```
BSR XKEYAL
```

```
BSR XKEYLC
```

```
BNE GEU68
```

→ fin (autre que lettre)

```
ADD.B D2, D3 ← ADDQ.B #1, D3
```

```
BRA GEU67
```

```
GEU68: MOVE.L D3, D0
```

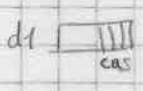
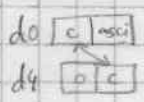
← ROL #8, D0

```
BRA GEU46
```

```
GEU680: MOVE #R80, D2
        CMP #4E, D4
        BEQ GEU65
        ADDQ #1, D2
        CMP #4A, D4
        BEQ GEU65
```

⊕ → \$280

⊖ → \$280



enter → return

TOUCHES
MOUVEMENTS
+ a, c et s

```
GEU69: CMP #72, D4
```

```
BNE GEU70
```

```
GEU70: MOVE #1C, D4
```

```
GEU70: BCLR #0, D2
```

```
BEQ GEU71
```

```
BSET #7, D4
```

si shifté: d4 = \$80 and Clavier

```
GEU71: LEA GEU73, A0
```

```
ASL #4, D2
```

```
ADD #100, D2
```

d2 = \$100 + \$20 a + \$40 c

```
GEU72: ADDQ #1, D2
```

```
MOVE.B (A0)+, D3
```

```
BEQ GEU74
```

→ non

```
CHP.B D3, D4
```

```
BEQ GEU65
```

```
BRA GEU72
```

```

GEU74: BSR XKEYLC
      BNE GEU740

```

```

(GEU74: CMP #6, D1
      BNE GEU740)

```

```

BSR XKEYLC
BNE GEU740

```

```

LEA GEU61F, A0
MOVE.B (A0, D2), D2

```

```

ADD # $180, D2
BRA GEU65

```

alt + control + lettre ?

\$180 + (0 à 25)

→ non

```

GEU740: SUBQ #1, D4 code d'avier

```

```

      BNE GEU742

```

```

      MOVEQ # $120, D2

```

```

      ADD D1, D2

```

```

      BRA GEU65

```

escape ?

\$120 + (0 à 5)

↓ oui



X


```
GEU742: MOVEQ #0, D2
```

⊗

```
: MOVE.B D0, D2
```

```
BNE GEU65
```

→ touches avec code ascii
met ascii = \$80 + code clavier français

```
MOVE D0, D2  
LSR #8, D2
```

```
CMPI #17, D2
```

```
BNE GEU75
```

```
MOVEQ #1A, D2
```

9I → 9A

```
GEU75: BSET #7, D2
```

```
BRA GEU65
```

```
BSR XKEYLC  
BNE GEM16  
LEA GEU61F, A0  
MOVE.B (A0, D2), D2
```

dze [0, 25] lettre A-Z du clavier
la lettre correspond à ce code

```
GEM16: MOVE D4, D2
```

```
OR #E0, D2
```

```
BRA GEU65
```