5 décode c$ = virchaine

↳ n° d4

d1 = min (x, y)
d3 = y − d1

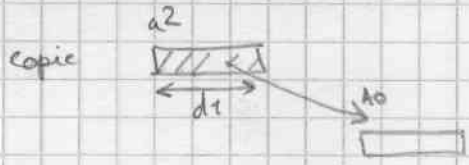
```
(SP) GAG40 : BSR   WNOMC      (cf) d2 = numéro
             MOVE  D2,-(SP)
             BSR   GAG42

             MOVE  (SP)+,D4

             BSR   LB95C
(SP) GAG400 :CMP.L D3,D1
             BLE   GAG41
             MOVE.L D3,D1
      GAG41 :SUB.L D1,D3
             RTS
```

```
(SP) GAG42 : BSR   DECCRE      (=)
             BNE   ERRIS7
             BSR   WCHAIN      (virchain)
             MOVE.L A0,A2
             MOVE.L D3,D1
             RTS
```

Copie disjointe



avance A0
détruit d0/d1/a2

(SP)
```
GAG43:MOVE.L  A6,-(SP)
      MOVE.L  A0,A6
      BSR     ILB76
      MOVE.L  (SP)+,A6
      RTS
```

met d3.L espaces en A0 et avance A0

(SP)
```
GAG44:MOVEQ   #32,d0
      BRA     GAG46
GAG45:MOVE.B  d0,(a0)+
GAG46:SUBQ.L  #1,d3
      BPL     GAG45
      RTS
```

5 lset c$=virchaine

rset c$=virchaine

```
YLSET: BSR   GAG40
       BSR   GAG43
       BSR   GAG44
       BRA   POPN

YRSET: BSR   GAG40
       BSR   GAG44
GAG47: BSR   GAG43
GAG48: BRA   POPN
```

mid$(c$, x [,y]) = virchaine ⑧ nom ⑨ $78 even YMIDC-Y YMIDC-Y

voir 52 pour fonctio.

```
YMIDJC:CMP.B  #"(",(A5)+   BSR  DECCRPGI
       BNE    ERRIS7
       BSR    WNOMC              c$      numéro d2
       BSR    DECCRVE            ⑨
       BSR    WADR               x = d3
       MOVE.L D2/D3,-(SP)        SUBQ.L #1,D3
       MOVEQ  #-1,d3             BMI    ERRRG
       BSR    WVGAD              [,y]    défaut = -1 (=∞)
       BSR    DECCRPDE           ")"
       BNE    ERRISY
       MOVE.L D3,-(SP)           y
       BSR    GAG42              = (A2 d1) virch
       MOVE.L (SP)+,D3
       CMP.L  D3,D1              d1 = min(d1, y)
       BCS    GAG49
       MOVE.L D3,D1
GAG49: MOVE.L (SP)+,D4           n° de c$
       BSR    LB95C              a0 ... d3
       ADD.L  (SP),A0
       SUB.L  (SP)+,D3
X      BMI    GAG48              → pas d'effet
       BSR    GAG400             d1 = min(d3,d1)
       BRA    GAG47              copie d1 octets
```