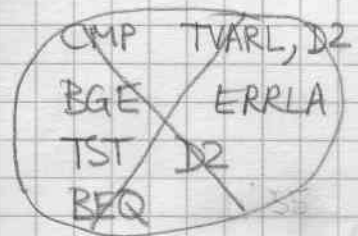


③ Cherche le nom du littéral # D2 ∈ [1, TVARV[  
 (si D5 = \$10)  
 variable  
 (si D5 = \$30)  
 chaîne  
 (si D5 = \$40)

→ met ⊙ [littéral phénix  
 sortie des factures premiers]

et l'écrit en A0<sup>e</sup>  
 avance A0  
 conserve le reste

XFLI1: MOVEM.L D0-D7/A1-A5, -(SP)



LEA TMIND, A2  
 MOVE #4020, D4 ——— index //

⊗ recherche aussi les littéraux cachés

PA50: ADDQ #4, A2  
 MOVE.L A2, D3  
 PA51: MOVE.L D3, A3  
 MOVE.L -(A3), D3

BEQ PA59  
 MOVE.L D3, A3  
 ADDQ #1, A3  
 MOVE.L A3, A1 ——— début du nom

PA52: TST.B (A3)+  
 BNE PA52  
 MOVE.B (A3), D0 ——— type  
 AND #F0, D0  
 CMP. D5, D0  
 BNE PA51 ——— ≠ { \$10 littéral

MOVE.B (A3)+, D0  
 MEVEN A3  
 MOVE (A3)+, D1  
 SUB D2, D1  
 BCS PA51  
 MOVE.L A3, A5

MOVE D0, D6

AND # \$F, D6

MOVEQ #1, D7

BRA PAS4

PAS3: MULL (A5)+, D7

PAS4: DBRA D6, PAS3

CMP <sup>D0</sup>~~D7~~, D1

BCC PAS1

MOVE D0, D2

PAS5: MOVE.B (A1)+, (A0)+

BNE PAS5

AND # \$F, D2

BEQ PAS8

MOVE.B # "(" , -1(A0)

SUBQ #1, D2

BSR XIND1

PAS6: MOVE (A1)+, D0

BSR XPREX

MOVE.B # ", " , (A0)+

DBRA D2, PAS6

MOVE.B # ")" , -1(A0)

PAS7: CLR.B (A0)+

PAS8: MOVEM.L (SP)+, D0-D7/A1-A5

RTS

PAS9: DBRA D4, PAS0

BRA ERRLA

nb d'indices  
← MOVE.L D0, -(SP)

D7 = nb de bits de cet indice

← MOVEM.L (SP)+, D0/D0

nb d'indices

→ zéro indices

met les indices sur A1

écrit D0.W en A0

x