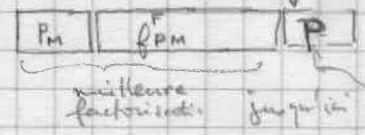
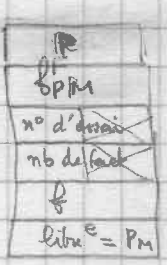
(10) Entrée $P_{A_0} = f(x) \in \mathbb{Z}[x]$ $cont'(f) = 1$
f sans facteur multiple

Sortie var_{A_2} = forme factorisée dans $\mathbb{Z}$ de f

```
XMFINT:MOVEQ #0,D0
       MOVEQ #-1,D1
       MOVEM.L  D0/D1/A0/A6,-(SP)
       |
       MOVE.L   A6,-(SP)    fPM
       MOVE.L   A6,-(SP)    P
       MOVE#$4002,(A6)+     p=1
```

1er passage $P_M = \emptyset$, $f_{PM} = \emptyset$, nb fact $= \infty$



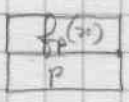



```
MJ48:MOVE.L  (SP),A0
     BSR     XMPRSA        nouveau pc
     MOVE.L  (SP),A0
     BSR     XLB76
     MOVE.L  (SP),A3       P
     MOVE.L  16(SP),A0     f
     ADDQ#8,A0             ldef(f)
     BSR     XMPOSE        conserve A3
     MOVE.L  A2,A6
     CMP     #$4000,(A2)
     BEQ     MJ48          → si nul changer P
     MOVE.L  16(SP),A0     f
     BSR     XMPNOR        pose conserve
```

recherche un nouveau $p > p_c$
remplace p_c par p_c suivant

vérif que $ldcoef(f) \not\equiv 0 \bmod p$

$f_{p_{A3}}(x) \equiv \lambda f(x) \bmod p$ normalisé




```
     MOVE.L  A2,A0
     BSR     XMPSQF        {détermine si fp(x) est squarefree
     EXG     A0,A6         conserve A0 A6
     BNE     MJ48
     EXG     A0,A6
```

(10)

```
MOVE.L  (SP),A3        P
MOVE.L  A0,-(SP)       fp(x)
CLR.B   XMFPLUF
BSR     XMFPLUF        fp(x)^F   (squaring)
```

$f_p(x)^F$ est-il le meilleur ?

```
MOVE    (A2),D0
CMP     16(SP),D0
BHI     MJ52           ⊗  → non
MOVE    D0,16(SP)          → oui, nouveau meilleur
MOVEM.L A2/A6,-(SP)
MOVEM.L (SP)+,D0/D1
MOVEM.L A2/A6,-(SP)
MOVE.L  D0,A6          ] P
MOVE.L  D1,A2          ] libre
MOVE.L  24(SP),A0
BSR     XLB76
MOVEM.L (SP)+,A2/A6
MOVE.L  A0,(SP)
BSR     XLB76
MOVE.L  A6,-(SP)
MOVE.L  20(SP),A0
BSR     XPOSE
MOVE.L  A6,-(SP)
MJ52:  MOVE.L  (SP)+,A6
ADDQ    #1,8(SP)       nb d'essais ± 1
MOVEM   8(SP),D0/D1/D2
```

$f_p^F(x)$

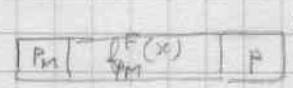
nouveau $f_p^F(x)_n$

Recopie

p

p_M

$p = p_M$



Doit-on continuer les essais de p ?

nb de fact
nb d'essais effectués

```
CMP     #2,D2
BNE     MJ54
MOVEM.L (SP)+,D1-D4/A0/A6
BRA     XPSAF
MJ54:  CMP     #5,D0
       BCS     MJ48
MOVE.L  (SP)+,A6
⊗ MOVE.L  (SP)+,A0
ADDQ    #8,SP
MOVEM.L (SP),A2/A3
⊗ MOVE.L  A0,-(SP)
```

cas irréductible

bidon

1
5

→

```
MJ54: SUB   #10,D2
      CMP   #4,D2
      BLE   MJ55
      MOVEQ #4,D2
MJ55: CMP   D0,D2
      BGE   MJ48
```

on effectue plusieurs essais de p si le nb de facteurs ≥ 11

11 : 2 essais
12 : 3 "
13 : 4 "
14 ou + : 5 "

→ change p

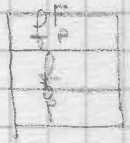
≤10 → -10 à 0
11 → 1
12 → 2
13 → 3
≥14 → 4

libre

(10)

Multiplie u_1 par ldcf(f) mod p

```
MOVE.L   A6,A5
⊗ LEA    8(A1),A0      pointe u1
LEA      8(A2),A4      pointe ldcf(f)
LEA      XMPCMULT,A2   (SP)   P_A0 → [A4] * P_A0 mod [A3]
⊗ BSR    XFREPI        ) remplace u1 par λu1
MOVE.L   (SP),A0
BSR      XLB76
MOVE.L   4(SP),A0
```



$f(x) = a_0 x^d + \dots + a_d$

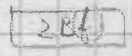
Calcul 2B(f)

$= 2^d \sqrt{(a_0)^2 + a_1^2 + \dots + a_d^2}$

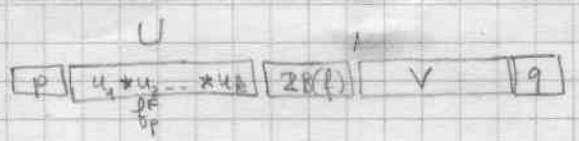
(coef des facteurs $\leq B(f)$)

```
MOVE.L   A6,-(SP)      S=0
MOVE     #$4000,(A6)+
ADDQ     #4,A0
M                      } k = nb de monomes - 1
MOVE.L   (A0)+,-(SP)   } di
```


```
MJ56 :MOVE.L  A0,-(SP)
      MOVE.L  A0,A1
      BSR     XMUL1          a_i^2
      MOVE.L  A2,A0
      MOVE.L  8(SP),A1       S
      BSR     XADD1          S + a_i^2
      MOVE.L  8(SP),A0       ) remplace S
      BSR     XLB76
      MOVE.L  (SP)+,A0
      BSR     SLN
      ADD     D0,A0
      ADDQ    #2,A0
      SUBQ    #1,(SP)
      BPL     MJ56
      MOVE.L  (SP)+,D2
      MOVE.L  (SP),A0
      ADD     D2,D2          2d
      BSR     XROT           2^ed Σ a_i^2
      MOVE.L  A2,A0
      BSR     SQRT1          int(2^d √(Σ a_i^2))
      MOVE.L  (SP),A0
      BSR     XLB76          2B(f)
```



(10)




```
MOVEM.L (SP), D0/A0/A1/A3        (A0 = δ^F_f, A1 = ≻, A3 = P)
BSR     XMFBE2                   met V
MOVE.L  A2,-(SP)
MOVE.L  A3,A0
BSR     XPOSE                    met q = P
MOVE.L  A2,-(SP)
```

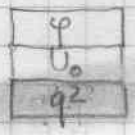
⊗

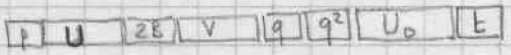
```
MJ58:MOVE.L  (SP),A0             q
     MOVE.L  8(SP),A1            2B(f)
     BSR     XCMP1
X    BCC     MJ68                → q ≥ 2B(f)
```

lift

⊗

```
MJ58:MOVE.L  (SP),A0
     MOVE.L  A0,A1
     BSR     XMUL1               q²
     MOVE.L  A2,-(SP)            ← MOVE.L A2,A3   q²
     MOVEM.L 12(SP), D5/A1/A4    (A1 = A0 = A_n U, A4 = f)   ⊗
     MOVE.L  D5,A5
⊗    LEA     8(A1),A0            pointe u_1
     ADDQ    #8,A4
     LEA     LIFTCU1,A2          } remplace dans u_1  ldcf(u_1) ≡ ldcf(f) mod q²
⊗    BSR     XFREPI                                   (A4)        (A3)
     MOVE.L  A2,-(SP)
     MOVE.L  A2,A0
     BSR     XDEVFP              pose P_{A2} = u_1 * u_2 --- * u_k = φ
     MOVE.L  A2,-(SP)            φ
     MOVE.L  8(SP),A3            q²
     MOVE.L  28(SP),A0           f
     MOVE.L  A2,A1               q
     BSR     XMPSUB              t = f - u_1 * -- * u_k  mod q²
     MOVE.L  A2,A0               t
     MOVE.L  12(SP),A1           q
     BSR     XDCTE               Pade P_{A2} = t/q
     MOVE.L  (SP),A0
     BSR     XLB76
```

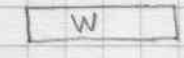


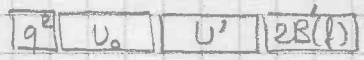


puis U'

```
                                  L U₀ x  q V
MOVEM.L (SP), A0/A1/A2/A3/A4
BSR     LIFTA                     varA2 =
MOVE.L  4(SP), A3                 U0
BSR     LIFTB                     varA2 = U'
MOVE.L  (SP), A0
BSR     XLB76
MOVE.L  20(SP), A0                2B(f)
BSR     XPOSE                     recopie 2B(f)
MOVE.L  A2, -(SP)
BSR     XPSP1                     par b=1
MOVE.L  A2, -(SP)

MOVE.L  24(SP), A4                V0
MOVE    (A4)+, D6                 k+1
ADDQ    #2, A4
MJ60 :SUBQ  #1, D6                          boucle sur i D6 = k, ... 1
X       BEQ     MJ66                        → fin
        MOVE.L  (A4)+, D0
        MOVE.L  A4, A0
        ADD.L   D0, A4
        BSR     XPSAP             pA2 = vi
        MOVE.L  8(SP), A3         U'i       ⊗
        MOVEM.L D6/A4, -(SP)
        MOVE    (A3)+, D6
        ADDQ    #2, A3                      boucle sur i D6 = k-1 1
MJ62:SUBQ  #1, D6
X       BEQ     MJ64
        MOVE.L  (A3)+, D0
        MOVE.L  A3, A1
        ADD.L   D0, A3
        CMP     2(SP), D6         ⊗
        BEQ     MJ62              → saute u'i
        MOVE.L  A2, A0
        MOVEM.L D6/A2/A3, -(SP)
        BSR     XMULP
        BSR     XLB76
```






```
MOVE.L  2(SP), A0   q²
MOVE.L  20(SP), A1  2Bf
BSR     XCMP1
BCC     MJ6$        → q² ≥ 2B(f)
```

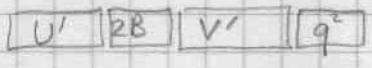
⊗

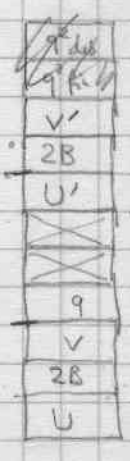
Calcul de $b(x) = 1 - \sum_{i=1}^{k} u'_1 * u'_2 \cdots \overset{v_i}{\not u'_i} * \cdots u'_k$

```
        MOVEM.L (SP)+,D6/A2/A3
        BRA     MJ62
MJ64:   MOVE.L  A2,A1
        MOVE.L  8(SP),A0        b
        BSR     XSUBP           b - u1' ? u1'
        BSR     XLB76
        MOVEM.L (SP)+,D6/A4
        BRA     MJ60
MJ66:   MOVE.L  (SP),A0         b
        MOVE.L  16(SP),A3       q²
        BSR     XMPPM           P_A2 ≡ b mod q²
        MOVE.L  A2,A0
        MOVE.L  20(SP),A1       q
        BSR     XDCTE           P_A2 = b/q = b'
        MOVE.L  A2,A0                P_A0 = b'
        MOVEM.L 12(SP),A1/A2/A3/A4
                U0  x   q   V

        BSR     LIFTA
        MOVE.L  24(SP),A3       V
        BSR     LIFTB           mt V'
        MOVE.L  (SP),A0
        BSR     XLB76

        MOVE.L  16(SP),A0       q²
        BSR     XPOSE
        MOVEM.L A2/A6,-(SP)
        MOVEM.L 12(SP),A2/A6
        EXG     A2,A6                ] U'
        MOVE.L  40(SP),A0
        BSR     XLB76                copie U'
        MOVE.L  A0,36(SP)
        MOVEM.L (SP)+
```





```
MOVE.L   A6,A5        fin q²
MOVE.L   A2,A4        q²
MOVE.L   (SP)+,A3     V'
MOVE.L   (SP)+,A6     2B'
MOVE.L   (SP)+,A2     U'
ADD  #20,SP
MOVE.L   (SP),A0      ancien U
BSR  XLB76            copie U'
MOVE.L   A6,-(SP)     nouveau 2B
MOVE.L   A3,A6
BSR  XLB76            copie 2B
MOVE.L   A6,-(SP)     nouveau V
MOVE.L   A4,A6
BSR  XLB76            copie V'
MOVE.L   A6,-(SP)
MOVE.L   A5,A6
BSR  XLB76            copie q²
BRA  MJ58
```

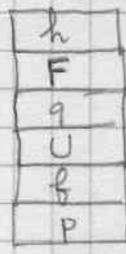
```
MJ68: MOVE.L   8(SP),A0
      BSR  XPOSE      q²
      MOVE.L   A6,A5  fin q²
      MOVE.L   A2,A6  fin U'
      MOVE.L   (SP)+,A2   U'
      ADD  #20,SP
      MOVE.L   (SP),A0    U ancien
      BSR  XLB76          copie U
      MOVE.L   A6,-(SP)
      MOVE.L   A5,A6
      BSR  XLB76          copie q²
```

[P][U][2B][V][q]

```
⊗  MJ68: MOVE.L  (SP)+,A2
         ADDQ  #4,SP
         MOVE.L  (SP),A0
         BSR  XLB76
```

→ MJ70

[P][U][q][F][h]

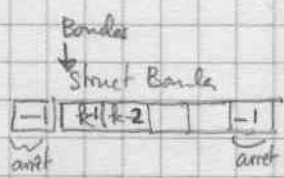



```
         MOVE.L  A6,-(SP)
         MOVE.L  #$14001,(A6)+      } F = 1  (facteurs trouvés)
         MOVE.L  12(SP),A0
         BSR  XPSAP                   h(x) = f(x)   partie à factoriser
         MOVE.L  A2,-(SP)

MJ70 :   MOVE.L  12(SP),A0            U = u1*...*uk
         MOVE  (A0)+,D6               k+1
         SUBQ  #2,D6       } ⊗       { SUBQ #8,SP  ⊗
x        BNE  MJ72         }         { MJ71: ADDQ #8,SP
                                      cas k = 1
         MOVE.L  (SP)+,A0     h(x)
         BSR  XPSAF           h(x)^F
         MOVE.L  A2,A1
         MOVE.L  (SP)+,A0     F
         BSR  XCONCP          F * h(x)^F
         ADD  #12,SP
         BRA  KL860           mis en libre

MJ72 :   CLR  -(SP)                   boucle sur m=1,... k-1 (D6)
MJ73 :   MOVE  (SP),D0        m-1
         ADDQ  #1,(SP)

MJ72 :   MOVE  D6,-(SP)
         CLR  -(SP)
MJ73 :   MOVEM  (SP),D0/D6    m-1     MOVE #-1,(A6)+
         ADDQ  #1,(SP)                MOVE.L A6,-(SP)
         MOVE.L  A6,-(SP)          MJ73: MOVE.L (SP)+,A6
MJ74 :   MOVE  D6,(A6)+
         SUBQ  #1,D6
         BMI  ERRFAT  MJ71  ⊗  → fin
         DBRA  D0,MJ74
         MOVE  D0,(A6)+   arrêt = (-1)
```



```
MJ76:MOVE.L  8(SP),A0
     ADDQ    #8,A0
     BSR     XPSAP
     ...
     MOVE.L  (SP),A3
     MOVE.L  A2,-(SP)
     MOVE.L  24(SP),A4
     MOVE    (A4)+,D6
     ADDQ    #2,A4
     ...
MJ78:MOVE    (A3)+,D5
     BMI     MJ88          → fin
MJ86:SUBQ    #1,D6
     MOVE.L  (A4)+,D0
     MOVE.L  A4,A1
     ADD.L   D0,A4
     CMP     D5,D6
     BNE     MJ86
     MOVE.L  (SP),A0       φ
     MOVEM.L D5/D6/A3/A4,-(SP)
     BSR     XMULP
     BSR     XLB76
     MOVEM.L (SP)+,D5/D6/A3/A4
     BRA     MJ78
MJ88:MOVE.L  (SP),A0       φ
     MOVE.L  20(SP),A3     q
     BSR     XMPPM
     MOVE.L  A2,A0
     BSR     XMPPB         φ balanced
     MOVE.L  A2,A0
     MOVE.L  A2,-(SP)
     BSR     XCONT
     MOVE.L  (SP)+,A0
     LEA     4(A2),A1
     BSR     XDCTE
     MOVE.L  (SP),A0
     BSR     XLB76
```

h(x)
pointe ldcf(h)

φ = ldcf(h)

boucles

$\varphi = ldcf(h) * u_{\alpha(1)} \cdots u_{\alpha(m)} = f$

U

```
MOVE.L  A6,-(SP)
CLR.L   (A6)+
BSR     XPOSE
MOVEM.L (SP)+,A2/A3
```

φ
boucle
m | k
R
F
q
U

⊗ BSR CALPRO ⊗

BSR XPP ⊗ p.p.
MOVE.L A2,A0

BSR XPP mal pp

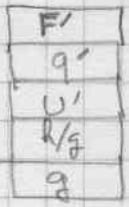
cont(φ)

$g(x) = pp(ldcf(h) * u_{\alpha(1)} \cdots u_{\alpha(m)} \underset{bal}{\bmod} q)$

```
        MOVE.L  (SP),A1         g(x)
        MOVE.L  12(SP),A0       h(x)
        BSR     XCTDIV1         division exacte ?
        BEQ     MJ94            → oui
                                ↓ non
        MOVE.L  (SP)+,A6
        LEA     -2(A6),A5       avance α(1) -- α(m)
        MOVE.L  A5,A0
MJ90 :  MOVE    -(A0),D0
        BMI     MJ73            → fin : augmenter m
MJ92    SUBQ    #1,D0
        BEQ     MJ90            ← MOVE.L A0,A1
        MOVE    D0,(A1)+
MJ92:   CMP.L   A5,A1
        BCC     MJ76            → nouveau sous-ensemble
        SUBQ    #1,D0
        BEQ     MJ90
        MOVE    D0,(A1)+
        BRA     MJ92

MJ94:   MOVE.L  A2,-(SP)                g(x) est un diviseur de h(x)
```


```
        MOVE.L  28(SP),A2       U
        MOVE.L  8(SP),A1        α
        MOVE    12(SP),D0       m
        MOVE.L  A6,-(SP)                nouveau U
        MOVE    (A2)+,D6
        MOVE    D6,(A6)
        SUB     D0,(A6)+
        MOVE    (A2)+,(A6)+             R'-m | $4001
MJ96:   SUBQ    #1,D6
        BEQ     MJ98            → fin
        CMP     (A1),D6
        BEQ     MJ97            → saute ui
        MOVE.L  (A2),D1         ↓ recopie ui
        ADDQ    #4,D1
        BSR     PLB76           ← ADDQ #2,A1   ⊗
        BRA     MJ96
MJ97:   ADD.L   (A2)+,A2
        BRA     MJ96
```




```
MJ98:MOVE.L  28(SP),A0      g
     BSR     XPOSE
     MOVE.L  A2,-(SP)
     MOVE.L  A6,-(SP)
     MOVE.L  16(SP),A0      g(x)
     BSR     XPSAF          g(x)^F
     MOVE.L  A2,A0                  )⊗
     MOVE.L  32(SP),A1      F       )
     BSR     XCONCP
     BSR     XLB76          F'
     MOVE.L  12(SP),A0      h/g
     BSR     XPSAP          recopie h/g = h'
     MOVE.L  A6,A5          h'
     MOVE.L  A2,A4          R'
     MOVE.L  (SP)+,A3       F'
     MOVE.L  (SP)+,A6       g'
     MOVE.L  (SP)+,A2       U'
     ADD     #28,SP
     MOVE.L  (SP),A0        U
     BSR     XLB76          copie U'
     MOVE.L  A6,-(SP)
     MOVE.L  A3,A6
     BSR     XLB76          copie g'
     MOVE.L  A6,-(SP)
     MOVE.L  A4,A6
     BSR     XLB76          copie F'
     MOVE.L  A6,-(SP)
     MOVE.L  A5,A6
     BSR     XLB76          copie h'
     BRA     MJ70
```