```
    |\^/|     Maple 9 (IBM INTEL LINUX)
._|\|   |/|_. Copyright (c) Maplesoft, a division of Waterloo Maple Inc. 2003
 \  MAPLE  /  All rights reserved. Maple is a trademark of
 <____ ____>  Waterloo Maple Inc.
      |       Type ? for help.
>  interface(screenwidth=120);
> with(LinearAlgebra):
# Rem plus lent que Powmod
>  Rem(X^100000,P,X) mod 101;
bytes used=4000260, alloc=3145152, time=0.07
bytes used=8000672, alloc=4324584, time=0.13
bytes used=12001156, alloc=4717728, time=0.21
bytes used=16001460, alloc=5635064, time=0.31
bytes used=42036900, alloc=24637024, time=1.11
                                                           0

> Powmod(X,100000,P,X) mod 101;
                                                           0

>  #Attention,  coeffs n’affiche pas  les coefficicients nuls
>  l:=[coeffs(X^4+2*X+1)];
                                                     l := [1, 2, 1]

>  op(l),seq(0,i=1..degree(P)-nops([l]));
                                                        1, 2, 1

>  # donc la methode precedente perd les zeros intermediaires.
>  # Il faut donc faire ainsi:
>  l:=[seq(coeff(X^4+2*X+1,X,i),i=0..degree(P)-1)];
                                                        l := [1]

>  L:=[]; L:=[op(L),1];L:=[op(L),2];L:=[op(L),4];
                                                        L := []

                                                        L := [1]

                                                      L := [1, 2]

                                                     L := [1, 2, 4]

>  #sum evalue le random avant!
>  sum(rand(0..20)()*X^i,i=0..n);
                                                      (n + 1)
                                                   6 X              6
                                                   ----------- - -----
                                                     X - 1      X - 1

>  randP:=n->add(rand(0..20)()*X^i,i=0..n-1)+X^n;
                                                                  i                     n
                               randP := n -> add(rand(0 .. 20)() X , i = 0 .. n - 1) + X

>  P:=expand(mul(randP(rand(1..7)()),i=1..5));
                                  2            3            4            5             6             7             8
P := 272646 + 763344 X + 1756206 X  + 3974994 X  + 6462501 X  + 9858434 X  + 14488029 X  + 17742874 X  + 20515135 X

                 9             10             11             12             13            14            15
     + 22842823 X  + 21808506 X   + 19666158 X   + 17202262 X   + 12663189 X   + 8572361 X   + 5691832 X

                16            17           18           19          20          21         22        23       24    25
     + 3107285 X   + 1397111 X   + 558632 X   + 191610 X   + 53863 X   + 12348 X   + 2306 X   + 322 X   + 28 X   + X

>  #Pour berlekamp, il ne faut pas de facteurs multiples.
>  gcd(P,diff(P,X));
                                                           1

>  #Attention {\’a} la bonne instruction pour trouver un noyau mod p, il y
>  #a 3 fonctions: nullspace de linalg, NullSpace de LinearAlgebra et
>  # enfin Nullspace qui ne depend d’aucun packages. c’est Nullspace
>  #qui marche bien mod p.
>  berl:=proc(p)
>  L:=[];
>  for i from 0 to degree(P) - 1 do
>  L:=[op(L),Vector(l)];
>  od:
>  F:=Matrix(L);Nullspace(F) mod p;
>  end proc;
berl := proc(p)
local L, i, F;
    L := []; for i from 0 to degree(P) - 1 do L := [op(L), Vector(l)] end do; F := Matrix(L); Nullspace(F) mod p
end proc

>  p:=2:L:=[];for i from 1 to 10 do ; p:=nextprime(p+1):L:=[op(L),Vector([nops(berl(nextprime(p))),p])];od:L;
                                                        L := []

bytes used=46038244, alloc=28044272, time=1.29
                               [24]  [24]  [24]  [24]  [24]  [24]  [24]  [24]  [24]  [24]
                              [[  ], [  ], [  ], [  ], [  ], [  ], [  ], [  ], [  ], [  ]]
                               [ 5]  [ 7]  [11]  [13]  [17]  [19]  [23]  [29]  [31]  [37]

>  #Non
>  #NB: lorsque l’on redemarre maple, parfois on a l’air d’avoir toujours le meme P
>  # pour qu’il y ait un peu de remontee, on prend un p<20
>  p:=17; Gcd(P,diff(P,X)) mod p;
                                                        p := 17

                                                           1

> N:=berl(p);LX:=Matrix(1,degree(P),(i,j)->X^(j-1));
      [16]  [16]  [16]  [16]  [16]  [16]  [16]  [16]  [16]  [16]  [16]  [16]  [16]  [16]  [16]  [16]  [16]  [16]  [16]
      [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]
      [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]
```

```
      [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]
      [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 1]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]
      [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]
      [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 1]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]
      [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]
      [ 0]  [ 0]  [ 0]  [ 0]  [ 1]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]
      [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]
      [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 1]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]
      [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]
      [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 1]
      [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]
      [ 0]  [ 0]  [ 0]  [ 1]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]
      [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]
      [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 1]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]
      [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]
      [ 0]  [ 0]  [ 1]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]
      [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]
      [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]
      [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]
      [ 0]  [ 1]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]
      [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]
N := {[ 0], [ 0], [ 0], [ 0], [ 0], [ 0], [ 0], [ 0], [ 0], [ 0], [ 0], [ 0], [ 0], [ 0], [ 0], [ 0], [ 1], [ 0],
      [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]
      [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 1]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]
      [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]
      [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]
      [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]
      [ 1]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]
      [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]
      [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 1]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]
      [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]
      [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]
      [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]
      [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 1]  [ 0]  [ 0]  [ 0]  [ 0]
      [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]
      [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 1]  [ 0]  [ 0]
      [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]
      [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 1]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]
      [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]
      [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 1]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]
      [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]
      [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 1]  [ 0]  [ 0]  [ 0]
      [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]
      [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]
      [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]  [  ]
      [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]  [ 0]

    [16]  [16]  [16]  [16]  [16]
    [  ]  [  ]  [  ]  [  ]  [  ]
    [ 1]  [ 0]  [ 0]  [ 0]  [ 0]
    [  ]  [  ]  [  ]  [  ]  [  ]
    [ 0]  [ 0]  [ 0]  [ 0]  [ 0]
    [  ]  [  ]  [  ]  [  ]  [  ]
    [ 0]  [ 0]  [ 0]  [ 0]  [ 0]
    [  ]  [  ]  [  ]  [  ]  [  ]
    [ 0]  [ 0]  [ 0]  [ 0]  [ 0]
    [  ]  [  ]  [  ]  [  ]  [  ]
    [ 0]  [ 0]  [ 0]  [ 0]  [ 0]
    [  ]  [  ]  [  ]  [  ]  [  ]
    [ 0]  [ 0]  [ 0]  [ 0]  [ 0]
    [  ]  [  ]  [  ]  [  ]  [  ]
    [ 0]  [ 0]  [ 0]  [ 0]  [ 0]
    [  ]  [  ]  [  ]  [  ]  [  ]
    [ 0]  [ 0]  [ 0]  [ 0]  [ 0]
    [  ]  [  ]  [  ]  [  ]  [  ]
    [ 0]  [ 0]  [ 0]  [ 0]  [ 0]
    [  ]  [  ]  [  ]  [  ]  [  ]
    [ 0]  [ 0]  [ 0]  [ 0]  [ 0]
    [  ]  [  ]  [  ]  [  ]  [  ]
    [ 0]  [ 0]  [ 0]  [ 0]  [ 0]
    [  ]  [  ]  [  ]  [  ]  [  ]
    [ 0], [ 0], [ 0], [ 0], [ 0]}
    [  ]  [  ]  [  ]  [  ]  [  ]
    [ 0]  [ 0]  [ 0]  [ 0]  [ 0]
    [  ]  [  ]  [  ]  [  ]  [  ]
    [ 0]  [ 0]  [ 1]  [ 0]  [ 0]
    [  ]  [  ]  [  ]  [  ]  [  ]
    [ 0]  [ 0]  [ 0]  [ 0]  [ 0]
    [  ]  [  ]  [  ]  [  ]  [  ]
    [ 0]  [ 0]  [ 0]  [ 0]  [ 0]
    [  ]  [  ]  [  ]  [  ]  [  ]
    [ 0]  [ 1]  [ 0]  [ 0]  [ 0]
    [  ]  [  ]  [  ]  [  ]  [  ]
    [ 0]  [ 0]  [ 0]  [ 0]  [ 0]
    [  ]  [  ]  [  ]  [  ]  [  ]
    [ 0]  [ 0]  [ 0]  [ 0]  [ 0]
    [  ]  [  ]  [  ]  [  ]  [  ]
    [ 0]  [ 0]  [ 0]  [ 0]  [ 0]
    [  ]  [  ]  [  ]  [  ]  [  ]
    [ 0]  [ 0]  [ 0]  [ 0]  [ 0]
    [  ]  [  ]  [  ]  [  ]  [  ]
    [ 0]  [ 0]  [ 0]  [ 0]  [ 0]
    [  ]  [  ]  [  ]  [  ]  [  ]
    [ 0]  [ 0]  [ 0]  [ 1]  [ 0]
    [  ]  [  ]  [  ]  [  ]  [  ]
    [ 0]  [ 0]  [ 0]  [ 0]  [ 1]

LX :=

    [         2    3    4    5    6    7    8    9    10    11    12    13    14    15    16    17    18    19
    [1 , X , X  , X  , X  , X  , X  , X  , X  , X   , X   , X   , X   , X   , X   , X   , X   , X   , X   ,
     21    22    23    24]
    X   , X   , X   , X  ]

> Q:=(LX.N[1])[1];
                                                                20
                                                     Q := 16 + X

#  verification:
> Rem(Powmod(Q,p,P,X)-Q mod p,P,X) mod p;
        2      3       4      5       6      7      8      9      10      11    12      13    14      15       16
11 X + X  + 2 X  + 10 X  + 8 X  + 10 X  + 6 X  + 7 X  + 2 X  + 5 X   + 3 X   + X   + 9 X   + X   + 3 X   + 11 X

          17       18      19       20       21       22      23       24
     + 7 X   + 14 X   + 6 X   + 15 X   + 14 X   + 13 X   + 8 X   + 10 X

# l’un des 3 pgcd est non trivial:
> Gcd(Q,P) mod p;
                                                         X + 13

> A:=Rem(Powmod(Q,(p-1)/2,P,X)-1 mod p,P,X) mod p;
         24       23      22    20      19       17       16      15      14       13      12       11       10    9
A := 11 X   + 10 X   + 2 X   + X   + 2 X   + 11 X   + 11 X   + 7 X   + 4 X   + 12 X   + 2 X   + 10 X   + 16 X   + X

           8      7       6      5      4      3      2
     + 10 X  + 8 X  + 16 X  + 2 X  + 9 X  + 2 X  + 9 X  + 5 X

> Gcd(A,P) mod p;
                                                           X

> B:=Rem(Powmod(Q,(p-1)/2,P,X)+1 mod p,P,X) mod p;
         24       23      22    20      19       17       16      15      14       13      12       11       10    9
B := 11 X   + 10 X   + 2 X   + X   + 2 X   + 11 X   + 11 X   + 7 X   + 4 X   + 12 X   + 2 X   + 10 X   + 16 X   + X

           8      7       6      5      4      3      2
     + 10 X  + 8 X  + 16 X  + 2 X  + 9 X  + 2 X  + 9 X  + 5 X + 2

> Gcd(B,P) mod p;
                                                         X + 3

>  unfacteur:=proc(d)
>  i:=1;
>  A:=1;B:=1;rep:=1;
>  #r:=rand(1..nops(N))();
>  while (i<nops(N) and degree(rep)=0 ) do
>  Q:=(LX.N[i])[1];
>  A:=Gcd(Q,d) mod p;
>  if degree(A)*(degree(A)-degree(d))<>0 then rep:=A ;
>  else A:=Rem(Powmod(Q,(p-1)/2,P,X)-1 mod p,P,X) mod p;
>  A:=Gcd(A,d) mod p;
>  if degree(A)*(degree(A)-degree(d))<>0 then rep:=A ;
>  else A:=Rem(Powmod(Q,(p-1)/2,P,X)+1 mod p,P,X) mod p;
>  A:=Gcd(A,d) mod p;
>  if degree(A)*(degree(A)-degree(d))<>0 then rep:=A fi;
>  fi;
>  fi;
>  i:=i+1;
>  od;
>  if degree(rep)=0 then d else rep  fi;
>  end proc;
unfacteur := proc(d)
local i, A, B, rep, Q;
    i := 1;
    A := 1;
    B := 1;
    rep := 1;
    while i < nops(N) and degree(rep) = 0 do
        Q := (LX . (N[i]))[1];
        A := Gcd(Q, d) mod p;
        if degree(A)*(degree(A) - degree(d)) <> 0 then rep := A
        else
            A := Rem((Powmod(Q, p/2 - 1/2, P, X) - 1) mod p, P, X) mod p;
            A := Gcd(A, d) mod p;
            if degree(A)*(degree(A) - degree(d)) <> 0 then rep := A
            else
                A := Rem((Powmod(Q, p/2 - 1/2, P, X) + 1) mod p, P, X) mod p;
                A := Gcd(A, d) mod p;
                if degree(A)*(degree(A) - degree(d)) <> 0 then rep := A end if
            end if
        end if;
        i := i + 1
    end do;
    if degree(rep) = 0 then d else rep end if
end proc

>  unfacteur(P);
                                                         X + 13

>  facteurpseudoirred:=proc(d)
>  t:=unfacteur(d);
>  tt:=d;
>  while (degree(t)<degree(tt)) do tt:=t;t:=unfacteur(t); od;
>  t;
>  end proc;
facteurpseudoirred := proc(d)
local t, tt;
    t := unfacteur(d); tt := d; while degree(t) < degree(tt) do tt := t; t := unfacteur(t) end do; t
end proc

>  facteurpseudoirred(P);
                                                         X + 13
```

```
> T:=P;a:=1;L:=[];
                                  2            3            4            5             6             7
T := 272646 + 763344 X + 1756206 X  + 3974994 X  + 6462501 X  + 9858434 X  + 14488029 X  + 17742874 X  + 20515135

                 9             10             11             12             13            14            15
     + 22842823 X  + 21808506 X   + 19666158 X   + 17202262 X   + 12663189 X   + 8572361 X   + 5691832 X

                16            17            18           19           20          21          22         23        24
     + 3107285 X   + 1397111 X   + 558632 X   + 191610 X   + 53863 X   + 12348 X   + 2306 X   + 322 X   + 28 X

                                                         a := 1

                                                        L := []

> while degree(T)>0 do T:=Quo(T,a,X) mod p;L:=[op(L),a];a:=facteurpseudoirred(T); od:L;
bytes used=50038444, alloc=28044272, time=1.43
bytes used=54039288, alloc=28044272, time=1.57
                       22       21      20      19    18      17      15      14    12       11       10       9
[1, X + 13, X, X + 3, X   + 12 X   + 6 X   + 8 X   + X   + 2 X   + 2 X   + 2 X   + X   + 10 X   + 13 X   + 10 X

           8       7      6      5      4      3       2
     + 10 X  + 16 X  + 8 X  + 7 X  + 9 X  + 8 X  + 12 X  + 8 X + 2]

#le nombre de facteurs doit etre la dim de ker F
>  if nops(N)=nops(L)-1 then print("on a bien trouve tous les facteurs") fi;
>  A:=L[2];B:=Quo(P,A,X) mod p; U:=’U’:V:=’V’:Gcdex(A,B,X,’U’,’V’) mod p:
                                                      A := X + 13

      24       23      22      21      20      19      18      17      16      15    14       13      12       11
B := X   + 15 X   + 8 X   + 9 X   + 8 X   + 5 X   + 6 X   + 2 X   + 8 X   + 6 X   + X   + 13 X   + 9 X   + 15 X

          10       9      8       7       6    5      4       3      2
     + 6 X   + 12 X  + 5 X  + 14 X  + 13 X  + X  + 2 X  + 10 X  + 9 X  + 6 X

# verification:
>  Rem(A*U+B*V,P,X) mod p;
                                                           1

#(A+p^iA’)(B+p^iB’)=P[p^(i+1)];A’B+B’A=(P-A.B)/p^i;A’=V*(P-AB)/p^i[p];
>  A:=L[2];B:=Quo(P,A,X) mod p; U:=’U’:V:=’V’:Gcdex(A,B,X,’U’,’V’) mod p:
                                                      A := X + 13

      24       23      22      21      20      19      18      17      16      15    14       13      12       11
B := X   + 15 X   + 8 X   + 9 X   + 8 X   + 5 X   + 6 X   + 2 X   + 8 X   + 6 X   + X   + 13 X   + 9 X   + 15 X

          10       9      8       7       6    5      4       3      2
     + 6 X   + 12 X  + 5 X  + 14 X  + 13 X  + X  + 2 X  + 10 X  + 9 X  + 6 X

>  # AU+BV=1[P];AB=P[p^i];(A+p^iAA)(B+p^iBB)=P[p^(i+1)]
>  for i from 1 to 1 do
>  PP:=expand((P-A*B)/p^i);AA:=Rem(PP*V,A,X) mod p;
>  BB:=Quo(expand(PP-B*AA),A,X) mod p;
>  A:=expand(A+p^i*AA);B:=expand(B+p^i*BB);
>  A;
>  od;
                                2           3           4           5           6            7            8
PP := 16038 + 44898 X + 103299 X  + 233815 X  + 380145 X  + 579907 X  + 852227 X  + 1043687 X  + 1206768 X

                9            10            11            12           13           14           15           16
     + 1343686 X  + 1282848 X   + 1156821 X   + 1011890 X   + 744883 X   + 504255 X   + 334809 X   + 182775 X

              17          18          19         20        21        22      23
     + 82181 X   + 32856 X   + 11267 X   + 3162 X   + 719 X   + 129 X   + 7 X

                                                        AA := 8

         23      22       21       20       19      18       17      16       15      14      12       11      10
BB := 9 X   + 8 X   + 12 X   + 15 X   + 13 X   + 8 X   + 13 X   + 5 X   + 15 X   + 6 X   + 9 X   + 14 X   + 9 X

           9      8       7      6    5       4      3      2
     + 16 X  + 8 X  + 15 X  + 5 X  + X  + 16 X  + 5 X  + 5 X  + 6 X + 11

                                                      A := X + 149

                       2       3        4       5       6        7        8        9        10        11        1
B := 187 + 108 X + 94 X  + 95 X  + 274 X  + 18 X  + 98 X  + 269 X  + 141 X  + 284 X  + 159 X   + 253 X   + 162 X

           13        14        15       16        17        18        19        20        21        22        23
     + 13 X   + 103 X   + 261 X   + 93 X   + 223 X   + 142 X   + 226 X   + 263 X   + 213 X   + 144 X   + 168 X

                                                        X + 149

>  remontee:=proc(AAA,BBB,P,j)
>  #pour minimiser la valeur absolue des coeff:
>  ‘mod‘:=modp;
>  A:=AAA;B:=BBB;
>  Gcdex(A,B,X,’U’,’V’) mod p:
>  # AU+BV=1[P];AB=P[p^i];(A+p^iAA)(B+p^iBB)=P[p^(i+1)]
>  for i from 1 to j do
>  PP:=expand((P-A*B)/p^i);AA:=Rem(PP*V,A,X) mod p;
>  BB:=Quo(expand(PP-B*AA),A,X) mod p;
>  A:=expand(A+p^i*AA) mod p^(i+1);B:=expand(B+p^i*BB) mod p^(i+1);
>  od;A,B;
>  end proc;
remontee := proc(AAA, BBB, P, j)
local A, B, i, PP, AA, BB;
    ‘mod‘ := modp;
    A := AAA;
    B := BBB;
    Gcdex(A, B, X, ’U’, ’V’) mod p;
    for i to j do
        PP := expand((P − A*B)/p^i);
        AA := Rem(PP*V, A, X) mod p;
        BB := Quo(expand(PP − B*AA), A, X) mod p;
        A := expand(A + p^i*AA) mod p^(i + 1);
        B := expand(B + p^i*BB) mod p^(i + 1)
    end do;
    A, B
end proc

>  A:=L[2];B:=Quo(P,A,X) mod p;remontee(A,B,P,10);
                                                      A := X + 13

      24       23      22      21      20      19      18      17      16      15    14       13      12       11
B := X   + 15 X   + 8 X   + 9 X   + 8 X   + 5 X   + 6 X   + 2 X   + 8 X   + 6 X   + X   + 13 X   + 9 X   + 15 X

          10       9      8       7       6    5      4       3      2
     + 6 X   + 12 X  + 5 X  + 14 X  + 13 X  + X  + 2 X  + 10 X  + 9 X  + 6 X

                                                                       2                  3                   4
X + 30075050515431, 33031919131645 X + 12079234310155 + 7883875250467 X  + 5587840674855 X  + 11414110672945 X

                       5                   6                   7                   8                   9
     + 19571595409843 X  + 15975640994585 X  + 13574251174363 X  + 24744709158563 X  + 11650900487257 X

                      10                  11                  12                 13                  14
     + 9211911998806 X   + 19505292184175 X   + 15036838313473 X   + 1887785045189 X   + 24029142616770 X

                       15                  16                  17                 18                 19
     + 18393356141054 X   + 33773194320658 X   + 27734907395812 X   + 6436525852759 X   + 5196423143528 X

                       20                  21                  22                 23    24
     + 15279639619766 X   + 19457654491701 X   + 19513227227299 X   + 4196845792230 X   + X

>  #Ne convient pas forcement, il faut alors essayer d’autres facteurs,
>  # ou bien des produits. Si la factorisation modulo $p$ a trop de
>  #facteurs c’est plus long.
>  for i from 2 to nops(L) do
>  A:=L[i];B:=Quo(P,A,X) mod p;S:=remontee(A,B,P,10)[1];
>  if abs(subs(X=0,S))<1000 then print("On pourrait essayer:",S);
>  if rem(P,A,X)=0 then print(A,"est un diviseur dans Z") fi;
>  fi;
>  od:
> borne:=proc(m,P)
> A:=Matrix([coeffs(P)]);
> binomial(m,floor(m/2))*evalf(sqrt((A.Transpose(A))[1,1]),5);
> end proc;
borne := proc(m, P)
local A;
    A := Matrix([coeffs(P)]); binomial(m, floor(1/2*m))*evalf(sqrt((A . (LinearAlgebra:−Transpose(A)))[1, 1]), 5)
end proc

> borne(5,P),evalf(p^10,3);
                                                           9          13
                                                0.553380 10 , 0.202 10

> quit
bytes used=57236172, alloc=28044272, time=1.64
```