```
    |\^/|     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);
>  Factor(X^8+1) mod 3;
                                                  4      2        4    2
                                                (X  + 2 X  + 2) (X  + X  + 2)

>  l:={1,2,3,4,3,4};
                                                      l := {1, 2, 3, 4}

>  l minus {1,3};
                                                            {2, 4}

>  l minus l;
                                                              {}

>  orbites:=proc(n)
>  if n mod 3 =0 then print("Erreur: 3 divise",n) fi;
>  l:={seq(i,i=0..n-1)};
>  j:=1;
>  while l<>{} do
>  i:=l[1];
>  o[j]:={i};k:=1;a:=0;
>  while a<>i do a:= 3^k*i mod n;o[j]:={op(o[j]),a};k:=k+1; l:=l od;
>  l:= l minus o[j]; j:=j+1:
>  od:seq(o[i],i=1..j-1);
>  end proc;
orbites := proc(n)
local l, j, i, o, k, a;
    if n mod 3 = 0 then print("Erreur: 3 divise", n) end if;
    l := {seq(i, i = 0 .. n - 1)};
    j := 1;
    while l <> {} do
        i := l[1];
        o[j] := {i};
        k := 1;
        a := 0;
        while a <> i do a := 3^k*i mod n; o[j] := {op(o[j]), a}; k := k + 1; l := l end do;
        l := l minus o[j];
        j := j + 1
    end do;
    seq(o[i], i = 1 .. j - 1)
end proc

>  Factor(X^32-1) mod 3;
      8      4         4      2                 2             2        2                  4    2                   8    4
    (X  + 2 X  + 2) (X  + 2 X  + 2) (X + 1) (X  + X + 2) (X  + 1) (X  + 2 X + 2) (X  + X  + 2) (X + 2) (X  + X  + 2)

>  orbites(32);
{0}, {1, 3, 9, 11, 17, 19, 25, 27}, {2, 6, 18, 22}, {4, 12}, {5, 7, 13, 15, 21, 23, 29, 31}, {8, 24}, {10, 14, 26, 30},

    {16}, {20, 28}

>  Factor(X^14-1) mod 3;
                  6      5    4      3    2               6    5    4    3    2
                (X  + 2 X  + X  + 2 X  + X  + 2 X + 1) (X  + X  + X  + X  + X  + X + 1) (X + 1) (X + 2)

>  orbites(14);
                                  {0}, {1, 3, 5, 9, 11, 13}, {2, 4, 6, 8, 10, 12}, {7}

>  #On remarque que pour tout i il y a autant d'orbites a i elements que de facteurs irreductibles de degre i
>  for i from 1 to 12 do nops(orbites(2^i)[2]),2^i end do;
                                                           1, 2

                                                           2, 4

                                                           2, 8

                                                          4, 16

                                                          8, 32

                                                          16, 64

                                                         32, 128

                                                         64, 256

                                                         128, 512

                                                        256, 1024

bytes used=4000516, alloc=3734868, time=0.06
                                                        512, 2048

bytes used=8003292, alloc=4259060, time=0.11
bytes used=12003800, alloc=5766112, time=0.16
bytes used=16008520, alloc=5766112, time=0.20
bytes used=20008988, alloc=5766112, time=0.25
                                                        1024, 4096

>  # On peut montrer que le noyau de la surjection donn\'ee par la
>  #reduction mod 4 est cyclique d'ordre 2^(n-2)
>  #-3=1+4.m ou m est impair, donc -3 est d'ordre maximal donc 3 aussi.
>  # En fait les elements d'ordre max sont ceux congrus a 5 ou -5 mod 8
>  for i from 1 to 12 do factor(X^(2^i)-1),2^i od;
                                                     (X - 1) (X + 1), 2
```

```
                                                 2
                                  (X - 1) (X + 1) (X  + 1), 4

                                              2        4
                              (X - 1) (X + 1) (X  + 1) (X  + 1), 8

                                          2        4        8
                         (X - 1) (X + 1) (X  + 1) (X  + 1) (X  + 1), 16

                                      2        4        8        16
                    (X - 1) (X + 1) (X  + 1) (X  + 1) (X  + 1) (X   + 1), 32

                                 2        4        8        16        32
               (X - 1) (X + 1) (X  + 1) (X  + 1) (X  + 1) (X   + 1) (X   + 1), 64

                            2        4        8        16        32        64
          (X - 1) (X + 1) (X  + 1) (X  + 1) (X  + 1) (X   + 1) (X   + 1) (X   + 1), 128

                       2        4        8        16        32        64        128
     (X - 1) (X + 1) (X  + 1) (X  + 1) (X  + 1) (X   + 1) (X   + 1) (X   + 1) (X    + 1), 256

                  2        4        8        16        32        64        128        256
(X - 1) (X + 1) (X  + 1) (X  + 1) (X  + 1) (X   + 1) (X   + 1) (X   + 1) (X    + 1) (X    + 1), 512

                    2        4        8        16        32        64        128        256        512
  (X - 1) (X + 1) (X  + 1) (X  + 1) (X  + 1) (X   + 1) (X   + 1) (X   + 1) (X    + 1) (X    + 1) (X    + 1), 10

                2        4        8        16        32        64        128        256        512        1024
(X - 1) (X + 1) (X  + 1) (X  + 1) (X  + 1) (X   + 1) (X   + 1) (X   + 1) (X    + 1) (X    + 1) (X    + 1) (X

    2048

                2        4        8        16        32        64        128        256        512        1024
(X - 1) (X + 1) (X  + 1) (X  + 1) (X  + 1) (X   + 1) (X   + 1) (X   + 1) (X    + 1) (X    + 1) (X    + 1) (X

     2048
    (X     + 1), 4096

>  #le poly cyclo phi_2^n est phi(n)
>  phi:=n->X^(2^(n-1))+1;
                                                      (n - 1)
                                                    (2       )
                                      phi := n -> X           + 1

>  for i from 1 to 10 do Factor(phi(i)) mod 3 od;
                                                X + 1

                                                 2
                                                X  + 1

                                         2             2
                                       (X  + X + 2) (X  + 2 X + 2)

                                         4    2        4      2
                                       (X  + X  + 2) (X  + 2 X  + 2)

                                         8    4        8      4
                                       (X  + X  + 2) (X  + 2 X  + 2)

                                         16    8        16      8
                                       (X   + X  + 2) (X   + 2 X  + 2)

                                         32    16        32      16
                                       (X   + X   + 2) (X   + 2 X   + 2)

                                         64    32        64      32
                                       (X   + X   + 2) (X   + 2 X   + 2)

                                         128      64        128    64
                                       (X    + 2 X   + 2) (X    + X   + 2)

bytes used=24012364, alloc=5766112, time=0.39
                                         256    128        256      128
                                       (X    + X    + 2) (X    + 2 X    + 2)

>  for i from 1 to 100 do if (3^i-1 mod 2^8) = 0  then print(i) fi; end
>  do;
                                                  64

>  #on prend i=64
>  Factor(X^128+1) mod 3;
                                         64    32        64      32
                                       (X   + X   + 2) (X   + 2 X   + 2)

>  P:=X^64+X^32-1;
                                                  64    32
                                           P := X   + X   - 1

>  `mod`:=mods;
                                               mod := mods

>  Factor(P) mod 3;
                                                64    32
                                               X   + X   - 1

>  #P convient
>  puissance:=(g,n)->Powmod(g,n,P,X) mod 3;
                              puissance := (g, n) -> Powmod(g, n, P, X) mod 3

>  q:=3^64;t:=(q-1)/2^8;
                                    q := 3433683820292512484657849089281
```

```
                    t := 1341282742301762689319472300 5
```

```
>  testcarre:=proc(g)
>  evalb(puissance(g,(q-1)/2)=1);
>  end proc;
          testcarre := proc(g) evalb(puissance(g, q/2 - 1/2) = 1) end proc

>  testcarre(1+X);
                                   true

>  # 1+X ne convient pas
>  testcarre(1+X^5);
                                   false

>  # 1+X^5 n'est pas un carre donc g est d'ordre 2^8.
>  g:=puissance(1+X^5,t);
                              g := X^63 - X^31

>  # verification:
>  b:='b';
                                  b := b

>  for i from 0 to 8 do b[i]:=puissance(g,2^i) od;
                             b[0] := X^63 - X^31

                               b[1] := -X^62

                            b[2] := -X^60 - X^28

                             b[3] := X^56 + X^24

                             b[4] := X^48 + X^16

                              b[5] := X^32 + 1

                              b[6] := X^32 - 1

                                b[7] := -1

                                b[8] := 1

>  inverse:=proc(x)
>  Gcdex(x,P,X,'invx','tutu') mod 3;
>  invx;
>  end proc;
      inverse := proc(x) Gcdex(x, P, X, 'invx', 'tutu') mod 3; invx end proc

>  z:=1+X;testcarre(z);
                                z := X + 1

                                   true

>  igcdex(t,2^8,'u','v');u,v;
                                    1

                   -107, 560614271196439874051498188 1

>  #dans cet exemple u est negatif, on cherche donc l'inverse de z
>  Gcdex(z,P,X,'invz','tutu') mod 3;
                                    1

>  #verification de l'isomorphisme  produit on doit retrouver z:
>  x:=puissance(inverse(z),-u*t);
                             x := -X^50 - X^18

>  y:=puissance(z,v*2^8);
                             y := -X^15 - X^14

>  Rem(x*y,P,X) mod 3;z;
                                  X + 1

                                  X + 1

>  # on verifie d'abord si q+1 est divisible par 4, si oui c'est tres simple.
>  q+1 mod 4;#tant pis..
                                    2

>  #racine de y:
>  raciney:=puissance(y,(t+1)/2);
raciney := X - X^45 - X^46 + X^47 - X^49 - X^50 + X^51 + X^52 - X^53 - X^54 + X^44 + X^33 - X^35 + X^36 - X^37 + X^39 + X^28 + X^60

     + X^55 + X^57 + X^59 - X^20 - X^21 + X^22 + X^15 - X^18 + X^5 - X^25 - X^26 + X^4 - X^24 - X^8 + X^56 - X^31 + X^63 + X^2 - X^16 + X^9

     + X^27 + X^30 + X^11 + X^12 + X^13 + X^3 + X^6 + X^32 - X^48

>  #verification:
>  puissance(raciney,2);y;
                                -X^15 - X^14

                                -X^15 - X^14

>  m:='m';xx:=x;#on sauve x
                                  m := m

                             xx := -X^50 - X^18

>  for i from 7 to 0 by -1 do if puissance(x,2^i)=-1 then
>  m[7-i]:=1;x:=Rem(x*inverse(b[7-i]),P,X) mod 3; else m[7-i]:=0;fi od;
>  # verification:
>  x:=1; for i from 0 to 7 do x:=Rem(x*puissance(b[i],m[i]),P,X) mod 3
                                  x := 1

>  od: x;xx;#on verife que l'on trouve bien la valeur sauvee.
                                -X^50 - X^18

                                -X^50 - X^18

>  racinex:=1;for i from 1 to 7 do racinex:=Rem(racinex*puissance(b[i-1],m[i]),P,X) mod 3
                               racinex := 1

>  od:puissance(racinex,2);x;
                                -X^50 - X^18

                                -X^50 - X^18

>  racinez:=Rem(racinex*raciney,P,X) mod 3;
racinez := 1 + X + X^43 + X^46 - X^51 - X^52 - X^53 + X^44 - X^33 + X^34 + X^36 - X^37 - X^39 - X^40 + X^41 - X^42 + X^28 + X^60

     + X^57 - X^59 + X^19 + X^15 - X^17 - X^18 - X^23 - X^25 - X^26 - X^4 - X^24 - X^63 - X^2 + X^16 - X^7 - X^10 - X^29 + X^30 + X

     - X^13 + X^6 + X^62 + X^32 + X^48

>  puissance(racinez,2);z;
                                  X + 1

                                  X + 1

> quit
bytes used=27448944, alloc=5766112, time=1.11
```