```
    |\^/|     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);
>  l:=ifactor(13*7*101);
                                                l :=  (7)  (13)  (101)

>  [seq(op(convert(op(l)[i],list)),i=1..nops(l))];
                                                     [7, 13, 101]

>  # ou bien avec
>  ifactors(13*7*101);
                                           [1, [[7, 1], [13, 1], [101, 1]]]

>  #ASTUCE, a savoir: 2 &^1000000000000000000 mod 3;
>  factor(x^2+1) mod 2;# factorise dans Z puis pqsse mod 2.
                                                         2
                                                        x  + 1

>  Factor(x^2+1) mod 2;# expression inerte donnee a mod qui factorise dans Z/2Z.
                                                              2
                                                       (x + 1)

>  alias(j=RootOf(x^2+x+1));
                                                          j

>  Factor(x^5+1) mod 2;
                                              4    3    2
                                            (x  + x  + x  + x + 1) (x + 1)

>  Factor(x^5+1,j) mod 2;
                                                 2                    2
                                       (x + 1) (x  + (j + 1) x + 1) (x  + j x + 1)

>  p:=3;
                                                        p := 3

>  n:=2; Factor(X^(p^n)-X) mod p;
                                                        n := 2

                                   2                  2                    2
                                 (X  + 1) (X + 1) X (X  + X + 2) (X + 2) (X  + 2 X + 2)

>  n:=6; Factor(X^(p^n)-X) mod p;
                                                        n := 6

  6    4      2        6      5    4      3    2              6      3            6      5
(X  + X  + 2 X  + 1) (X  + 2 X  + X  + 2 X  + X  + 2 X + 1) (X  + 2 X  + X + 1) (X  + 2 X  + 2 X + 2)

      6      5    4      3      2            6    4    3        6    5    4      3            6    2
    (X  + 2 X  + X  + 2 X  + 2 X  + X + 2) (X  + X  + X  + 1) (X  + X  + X  + 2 X  + X + 1) (X  + X  + X + 1)

      6      5      4    2            6      5    4    3              2                6    4      3
    (X  + 2 X  + 2 X  + X  + X + 1) (X  + 2 X  + X  + X  + 2 X + 1) (X  + 1) (X + 1) (X  + X  + 2 X  + X + 2)

      3              6      5      4    3        6      4    3    2              6      5      4      2
    (X  + 2 X + 1) (X  + 2 X  + 2 X  + X  + 2) (X  + 2 X  + X  + X  + X + 2) X (X  + 2 X  + 2 X  + 2 X  + X + 2)

      6      5    4    3      2        6    5    3    2        6      5    4    2
    (X  + 2 X  + X  + X  + 2 X  + 1) (X  + X  + X  + X  + 1) (X  + 2 X  + X  + X  + X + 2)

      6    5    4    3    2            6      5      4      3    2            6    4      3    2
    (X  + X  + X  + X  + X  + X + 2) (X  + 2 X  + 2 X  + 2 X  + X  + X + 2) (X  + X  + 2 X  + X  + X + 2)

      6      4    2              6      5      4    3      2            6    5      4    3    2
    (X  + 2 X  + X  + 2 X + 2) (X  + 2 X  + 2 X  + X  + 2 X  + X + 1) (X  + X  + 2 X  + X  + X  + 2 X + 2)

      6      5      4      3    2        6    5        6    5    4    2              6    5    4
    (X  + 2 X  + 2 X  + 2 X  + X  + 2) (X  + X  + 2) (X  + X  + X  + X  + 2 X + 2) (X  + X  + X  + 1)

      6      5      4      3      2            6    5    4    3              6      5      4    3      2
    (X  + 2 X  + 2 X  + 2 X  + 2 X  + X + 1) (X  + X  + X  + X  + 2 X + 1) (X  + 2 X  + 2 X  + X  + 2 X  + 2)

      6      4      3    2        6      4    3              6      4    3    2              6    4      2
    (X  + 2 X  + 2 X  + X  + 2) (X  + 2 X  + X  + 2 X + 1) (X  + 2 X  + X  + X  + 2 X + 1) (X  + X  + 2 X  + 2 X + 2)

      6    5      4      3      2        6      3              6      5    3        6      3    2
    (X  + X  + 2 X  + 2 X  + 2 X  + 2) (X  + 2 X  + 2 X + 2) (X  + 2 X  + X  + 1) (X  + 2 X  + X  + 1)

      6    3            6      5      3        6      5    4        3      2              6
    (X  + X  + X + 2) (X  + 2 X  + 2 X  + 2) (X  + 2 X  + X  + 1) (X  + 2 X  + 2 X + 2) (X  + X + 2)

      6      4      3            6    5      4    3      2              6    5      4      2
    (X  + 2 X  + 2 X  + X + 1) (X  + X  + 2 X  + X  + 2 X  + 2 X + 1) (X  + X  + 2 X  + 2 X  + 2 X + 2)

      6      5    4      3    2              6    5    3            6      4        6    5      4    3    2
    (X  + 2 X  + X  + 2 X  + X  + 2 X + 2) (X  + X  + X  + X + 1) (X  + 2 X  + 1) (X  + X  + 2 X  + X  + X  + 2)

      3      2        6    5    3      2            6      4      3    2            2
    (X  + 2 X  + 1) (X  + X  + X  + 2 X  + X + 2) (X  + 2 X  + 2 X  + X  + X + 1) (X  + X + 2)

      6      3      2            6    5    4    3            6    5      4      3        6      3    2
    (X  + 2 X  + 2 X  + X + 1) (X  + X  + X  + X  + X + 2) (X  + X  + 2 X  + 2 X  + 1) (X  + 2 X  + X  + X + 2)

      6    3      2              6    5      4    2        6    3    2              6    5    4    3      2
    (X  + X  + 2 X  + 2 X + 1) (X  + X  + 2 X  + X  + 2) (X  + X  + X  + 2 X + 2) (X  + X  + X  + X  + 2 X  + 2 X + 2)

      6    5    4            6    4    3              6      5      3    2        6      5    2
    (X  + X  + X  + X + 1) (X  + X  + X  + 2 X + 2) (X  + 2 X  + 2 X  + X  + 1) (X  + 2 X  + X  + 2 X + 1)
```

```
      6    5    4      2              6    5      4      3    2              6      5    4
    (X  + X  + X  + 2 X  + 2 X + 1) (X  + X  + 2 X  + 2 X  + X  + 2 X + 2) (X  + 2 X  + X  + 2 X + 1)

      6      5      4    2        6    5    4      2        3              6      5    3    2
    (X  + 2 X  + 2 X  + X  + 2) (X  + X  + X  + 2 X  + 2) (X  + 2 X + 2) (X  + 2 X  + X  + X  + X + 1)

      6    5      3    2              6    5      4    3            6    4      3              6    5
    (X  + X  + 2 X  + X  + 2 X + 1) (X  + X  + 2 X  + X  + X + 2) (X  + X  + 2 X  + 2 X + 1) (X  + X  + X + 2)

      6      5    4      2            6      4      3    2              3    2
    (X  + 2 X  + X  + 2 X  + X + 1) (X  + 2 X  + 2 X  + X  + 2 X + 2) (X  + X  + 2 X + 1)

      6      5    4      3                      6      5      4      3      2
    (X  + 2 X  + X  + 2 X  + 2 X + 2) (X + 2) (X  + 2 X  + 2 X  + 2 X  + 2 X  + 2 X + 2)

      6      5    4    3      2            6    5      3        6      5      4      3              6    3    2
    (X  + 2 X  + X  + X  + 2 X  + X + 2) (X  + X  + 2 X  + 1) (X  + 2 X  + 2 X  + 2 X  + 2 X + 2) (X  + X  + X  +

      6      5      3              6    4      3        6      5      4    3        6    5    3
    (X  + 2 X  + 2 X  + 2 X + 1) (X  + X  + 2 X  + 1) (X  + 2 X  + 2 X  + X  + 1) (X  + X  + X  + 2)

      6    2              6      5      3      2              6      5    3      2        6      5        2
    (X  + X  + 2 X + 1) (X  + 2 X  + 2 X  + 2 X  + 2 X + 2) (X  + 2 X  + X  + 2 X  + 2) (X  + 2 X  + 2) (X  + 2 X

      6      5      3    2              6    4    3            6    5    2            6    5      3      2
    (X  + 2 X  + 2 X  + X  + 2 X + 2) (X  + X  + X  + X + 1) (X  + X  + X  + X + 1) (X  + X  + 2 X  + 2 X  + 2)

      6    5    3    2            6    5      3      2        6      5    4      3
    (X  + X  + X  + X  + X + 2) (X  + X  + 2 X  + 2 X  + 1) (X  + 2 X  + X  + 2 X  + X + 1)

      6    4    3    2              6      4    2            3    2            6    5      4    2
    (X  + X  + X  + X  + 2 X + 2) (X  + 2 X  + X  + X + 2) (X  + X  + X + 2) (X  + X  + 2 X  + X  + 2 X + 1)

      6      5    4      2        6    4      2            6              6      5    3      2
    (X  + 2 X  + X  + 2 X  + 2) (X  + X  + 2 X  + X + 2) (X  + 2 X + 2) (X  + 2 X  + X  + 2 X  + 1)

      6    5    4      3      2        6    5    4    3    2            6      5    3    2
    (X  + X  + X  + 2 X  + 2 X  + 1) (X  + X  + X  + X  + X  + X + 1) (X  + 2 X  + X  + X  + 2 X + 1)

      6    5      4      3        6      4    2        6    5    4      3      2              3    2
    (X  + X  + 2 X  + 2 X  + 2) (X  + 2 X  + X  + 1) (X  + X  + X  + 2 X  + 2 X  + 2 X + 2) (X  + X  + 2)

      6    5      4      3      2              6    5      4    3      2            3      2
    (X  + X  + 2 X  + 2 X  + 2 X  + 2 X + 1) (X  + X  + 2 X  + X  + 2 X  + X + 2) (X  + 2 X  + X + 1)

      6    5      3    2            6      4    3    2        6      5      4    3    2            6      2
    (X  + X  + 2 X  + X  + X + 1) (X  + 2 X  + X  + X  + 2) (X  + 2 X  + 2 X  + X  + X  + X + 2) (X  + 2 X  + 1)

      6    3
    (X  + X  + 2 X + 1)

>  p:=107;N:=2^p-1;isprime(N);
                                                  p := 107

                                    N := 162259276829213363391578010288127

                                                    true

>  p:=nextprime(200);N:=2^p-1;isprime(N);
                                                  p := 211

                   N := 3291009114642412084309938365114701009965471731267159726697218047

                                                   false

>  # ifactor(N);
>  ifactor(N-1);
          2     2
 (2)  (3)   (7)   (11)  (31)  (43)  (71)  (127)  (151)  (211)  (281)  (331)  (337)  (86171)  (122921)  (152041)

     (29191)  (106681)  (664441)  (5419)  (1564921)

>  for i from 1 to 100 do p:=ithprime(i):N:=2^p-1:
>  if isprime(N) then print(p) fi;  od:
                                                      2

                                                      3

                                                      5

                                                      7

                                                      13

                                                      17

                                                      19

                                                      31

                                                      61

                                                      89

                                                     107

                                                     127

                                                     521
```

```
>  # on prend 521, alors
>  isprime(2^521−1);
                                                         true

>  # ifactor(2^521−2); # tres long. Donc c’est plutot une reponse
>  # probabiliste qu’exacte.
>  p:=107;N:=2^p−1;l:=ifactor(N−1);
                                                       p := 107

                                        N := 162259276829213363391578010288127

                          l :=  (2)  (3)  (107)  (28059810762433)  (69431)  (20394401)  (6361)

>  L:=[seq(op(convert(op(l)[i],list)),i=1..nops(l))];
                                L := [2, 3, 107, 28059810762433, 69431, 20394401, 6361]

>  cherchex:=proc(n,N)
>  f:=rand(1..N−1);
>  x:=2;
>  while (x&^n mod N =1) or (x&^(N−1) mod N<>1) do x:=f() od;
>  x;
>  end proc;
cherchex := proc(n, N)
local f, x;
    f := rand(1 .. N − 1); x := 2; while ‘&^‘(x, n) mod N = 1 or ‘&^‘(x, N − 1) mod N <> 1 do x := f() end do; x
end proc

>  for i from 1 to nops(L) do n:=(N−1)/L[i]; cherchex(n,N)
>  od;print(N,"est certifie premier");
                                        n := 81129638414606681695789005144063

                                           28733913173111519853864534149814

                                        n := 54086425609737787797192670096042

                                          13453666859701088878580069192636 7

                                         n := 1516441839525358536369887946618

                                                         2

                                              n := 5782621921543716222

                                           97144550367941533627736278853100

                                          n := 2336986026835467779400815346

                                          16091965585362053174593081183632 0

                                            n := 7956069748222238220753726

                                          16192267194312861472890189529676 3

                                          n := 2550845414702300949403836036 6

                                           86174533699022543166210851601668

                               162259276829213363391578010288127, "est certifie premier"

>  p:=101; N:=2^p−1;
                                                       p := 101

                                         N := 2535301200456458802993406410751

>  ‘mod‘:=mods; # pour remettre entre 0 et N: ‘mod‘:=modp;
                                                     mod := mods

>  a:=rand(N)();Power(a,(N−1)/2) mod N;
                                         a := 1507861968677516949887562507311

                                              2669085548001623074719798767 75

>  # ou a&^((N−1)/2) mod N;
> member(1,{−1,1});member(2,{−1,1});
                                                         true

                                                        false

>  testeuler:=proc(N)
>  f:=rand(0..N−1);
>  ‘mod‘:=mods;
>  a:=f();i:=0;
>  while (member(Power(a,(N−1)/2) mod N,{−1,1})) and i<100 do a:=f();i:=i+1;od ;
>  if i<100 then false #print(a,"ne passe pas le test d’euler")
>  else true #print("Je ne sais pas")
>  fi;
>  end proc;
testeuler := proc(N)
local f, a, i;
    f := rand(0 .. N − 1);
    ‘mod‘ := mods;
    a := f();
    i := 0;
    while member(Power(a, 1/2*N − 1/2) mod N, {−1, 1}) and i < 100 do a := f(); i := i + 1 end do;
    if i < 100 then false else true end if
end proc

>  p:=100;for i from 1 to 100 do p:=nextprime(p);N:=2^p−1:if testeuler(N)
                                                       p := 100

>  then print("Je ne sais pas pour p=",p) fi;od:
```

```
                                             "Je ne sais pas pour p=", 107

                                             "Je ne sais pas pour p=", 127

bytes used=4000624, alloc=2817532, time=0.82
                                             "Je ne sais pas pour p=", 521

                                             "Je ne sais pas pour p=", 607

>  #rep 107,127,521,607
> quit
bytes used=5603132, alloc=2817532, time=1.62
```