```
    |\^/|     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);
>  interface(screenwidth=120);
#Calculer pour $n\in \nat$: $u_nv_{n-1}-u_{n-1}v_n$. Montrer que pour tout $\alpha$ on a
#$[a_0,a_1,...,a_{n-1},\alpha]=\frac{\alpha u_{n-1}+u_{n-2}}{\alpha v_{n-1}+v_{n-2}}$
# En d{\'e}duire lorsque les $a_n$ sont dans $\nat^*$ que \frac{u_n}{v_n}$ est
# la fraction irr{\'e}ductible correspondant {\'a} $[a_0,a_1,\dots,a_n]$.
>  #On fait une proc{\'e}dure facteurs qui donne les facteurs d'un entier
>  facteurs:=proc(n)
>  l:=ifactors(n);
>  {seq(i[1],i=l[2])};
>  end proc;
                    facteurs := proc(n) local l; l := ifactors(n); {seq(i[1], i = l[2])} end proc

#on retourne un ensemble plutot qu'une liste pour ne pas repeter de facteurs.
>  facteurs(123456789); {1,2} union {1,3};
                                                    {3, 3607, 3803}

                                                       {1, 2, 3}

>  facteursliste:=proc(a)
>  l:={};
>  for i in a do l:=l union facteurs(i); od;
>  end proc;
         facteursliste := proc(a) local l, i; l := {}; for i in a do l := l union facteurs(i) end do end proc

# on aurait aussi pu utiliser factorset apres avoir fait with(numtheory);
>  facteursliste([123456789,5*7,7*13]);#on teste
                                               {3, 5, 7, 13, 3607, 3803}

>  #pour a utiliser mods et non modp. recherche d'une assez bonne liste pour n
>  n:=nextprime(100)*nextprime(200);B:=floor(evalf(sqrt(n)))+1;
                                                       n := 21311

                                                        B := 146

>  #exemple a la main:
>  for i from 0 to 6 do ifactor(mods((B+i)^2,n)); od;
                                                           (5)

                                                       (2)  (149)

                                                          (593)

                                                     (2)  (5)  (89)

                                                       (29)  (41)

                                                     (2)  (5)  (149)

                                                       (11)  (163)

>  #on voit  une relation entre i=0,1,5
>  P:={-1,2,5,149};bb:=[B,B+1,B+5];
                                                   P := {-1, 2, 5, 149}

                                                  bb := [146, 147, 151]

> a:=[seq(mods(bb[i]^2,n),i=1..3)];
                                                   a := [5, 298, 1490]

>  c:=P[2]*P[3]*P[4];b:=bb[1]*bb[2]*bb[3];
                                                        c := 1490

                                                      b := 3240762

>  igcd(c-b,n);#rate! on a b=c [n]
                                                         21311

>  # on peut aussi chercher autour de sqrt(k*n)
>  B:=floor(evalf(sqrt(3*n)))+1;
                                                        B := 253

>  for i from 0 to 6 do ifactor(mods((B+i)^2,n)); od;
                                                          2
                                                       (2)   (19)

                                                       (11)  (53)

                                                      2
                                                   (2)   (3)  (7)  (13)

                                                       (7)  (229)

                                                          2      2
                                                       (2)   (23)

                                                       (3)  (877)

                                                          2
                                                       (2)   (787)

>  #i=4 est un carre, c'est particulierement favorable.
>  c:=sqrt(mods((B+4)^2,n));b:=B+4;igcd(c-b,n);#OK 211 divise n
                                                        c := 46

                                                        b := 257
```

```
                                                          211

>  #un peu moins particulier:
>  B:=floor(evalf(sqrt(4*n)))+1;
                                                        B := 292

>  for i from 0 to 6 do ifactor(mods((B+i)^2,n)); od;
                                                          2
                                                       (2)   (5)

                                                              2
                                                      (5)  (11)

                                                         3
                                                      (2)   (149)

                                                      (13)  (137)

                                                         2
                                                      (2)   (593)

                                                      (5)  (593)

                                                        3
                                                    (2)   (5)  (89)

>  #le produit des 2 premiers est un carre.
>  c:=2*5*11;b:=B*(B+1);igcd(b-c,n);
                                                        c := 110

                                                       b := 85556

                                                          101

>  #On automatise ce que l'on vient de faire.
>  a:={};b:={};for i from 1 to 10 do if max(op(facteurs(mods((B+i)^2,n))))<500
                                                         a := {}

                                                         b := {}

>  then a:=a union {mods((B+i)^2,n)}; b:=b union {B+i}; fi; od;
>  P:={-1} union facteursliste(a);P:=[op(P)];a:=[op(a)];b:=[op(b)];
                                       P := {-1, 2, 5, 11, 13, 29, 41, 89, 137, 149, 487}

                                       P := [-1, 2, 5, 11, 13, 29, 41, 89, 137, 149, 487]

                                         a := [605, 1192, 1781, 3560, 4756, 5357, 5960]

                                            b := [293, 294, 295, 298, 300, 301, 302]

>  maxpow:=proc(p,n)
>  nn:=n;k:=0;
>  if p=-1 then k:=(1-sign(n))/2; fi;
>  while (nn mod p = 0)and (p>1) do nn:=nn/p; k:=k+1; od; k; end proc;
maxpow := proc(p, n)
local nn, k;
    nn := n;
    k := 0;
    if p = -1 then k := 1/2 - 1/2*sign(n) end if;
    while nn mod p = 0 and 1 < p do nn := nn/p; k := k + 1 end do;
    k
end proc

>  maxpow(5,5^3*7*2);
                                                           3

>  alpha:=proc(a,P)
>  Matrix([seq([seq(maxpow(pj,ai),pj=P)],ai=a)]);
>  end proc;
                  alpha := proc(a, P) Matrix([seq([seq(maxpow(pj, ai), pj = P)], ai = a)]) end proc

>  #Attention: il faut trouver la bonne instruction pour trouver un noyau mod 2
>  with(LinearAlgebra):
>  Nullspace(Matrix([[2]])) mod 2;
                                                         {[1]}

>  A:=alpha(a,P);V:=(Nullspace(Transpose(A)) mod 2);
                                      [0    0    1    2    0    0    0    0    0    0    0]
                                      [                                                  ]
                                      [0    3    0    0    0    0    0    0    0    1    0]
                                      [                                                  ]
                                      [0    0    0    0    1    0    0    0    1    0    0]
                                      [                                                  ]
                                 A := [0    3    1    0    0    0    0    1    0    0    0]
                                      [                                                  ]
                                      [0    2    0    0    0    1    1    0    0    0    0]
                                      [                                                  ]
                                      [0    0    0    1    0    0    0    0    0    0    1]
                                      [                                                  ]
                                      [0    3    1    0    0    0    0    0    0    1    0]

                                                          [1]
                                                          [ ]
                                                          [1]
                                                          [ ]
                                                          [0]
                                                          [ ]
                                                     V := {[0]}
                                                          [ ]
                                                          [0]
```

```
                                                       [ ]
                                                       [0]
                                                       [ ]
                                                       [1]

>  c:=1;bb:=1; for i from 1 to nops(a) do if (V[1])[i]=1 then
                                                    c := 1

                                                   bb := 1

>  c:=c*a[i];bb:=bb*b[i]; fi ;od;c:=sqrt(c); igcd(bb-c,n);
                                                  c := 65560

                                                     101

>  # on y incorpore des fractions continues. Attention a la precision:
>  with(numtheory):
>  cfrac(sqrt(n),7);
                                                       1
                                      145 + -----------------------------------
                                                         1
                                            1 + -------------------------------
                                                            1
                                                57 + --------------------------
                                                              1
                                                     2 + ----------------------
                                                                1
                                                         1 + ------------------
                                                                  1
                                                             1 + --------------
                                                                      1
                                                                 11 + --------
                                                                      12 + ...

>  #si l'on travaille avec 10 chiffres,...
>  x:=evalf(sqrt(n),10):B:=floor(x):a:=[B]:
>  for i from 1 to 7 do
>   x:=evalf((1/(x-B),10)):B:=floor(x):
>   a:=[op(a),B] od:a;#...le 7 ieme chiffre est deja faux.
                                      [145, 1, 57, 2, 1, 1, 10, 1]

>  # On travaille avec 100 chiffres.
>   x:=evalf(sqrt(n),100);
      x := 145.9828757080774868238013282492534017464963318593074196579226838322290491036498759065439710443188176

>   a:={}:b:={}:uu:=0:u:=1:vv:=1:v:=0:B:=floor(x):
#On remarque que  u*vv-v*uu vaut +/- 1
>   for i from 1 to 7 do
>  oldu:=u:u:=B*u+uu:uu:=oldu:oldv:=v:v:=B*v+vv:vv:=oldv:
>  a:=a union {mods((u)^2,n)}: b:=b union {u}:
>  x:=evalf((1/(x-B),100)):B:=floor(x):print("u*vv-v*uu",u*vv-v*uu);
>  od: evalf(u/v,100);#on verifie que ca tend vers racine  de n
                                                  "u*vv-v*uu", -1

                                                   "u*vv-v*uu", 1

                                                  "u*vv-v*uu", -1

                                                   "u*vv-v*uu", 1

                                                  "u*vv-v*uu", -1

                                                   "u*vv-v*uu", 1

                                                  "u*vv-v*uu", -1

        145.9828757012105107764983761440803070563920873929731325656923531148509005019191024505462060820785356

>  P:={-1} union facteursliste(a):P:=[op(P)];a:=[op(a)];b:=[op(b)];
                                       P := [-1, 2, 5, 11, 13, 23, 83]

                                   a := [-286, -166, -115, -23, 5, 25, 121]

                              b := [145, 146, 8467, 17080, 25547, 42627, 494444]

>  A:=alpha(a,P);
                                         [1    1    0    1    1    0    0]
                                         [                               ]
                                         [1    1    0    0    0    0    1]
                                         [                               ]
                                         [1    0    1    0    0    1    0]
                                         [                               ]
                                    A := [1    0    0    0    0    1    0]
                                         [                               ]
                                         [0    0    1    0    0    0    0]
                                         [                               ]
                                         [0    0    2    0    0    0    0]
                                         [                               ]
                                         [0    0    0    2    0    0    0]

> V:=(Nullspace(Transpose(A)) mod 2);
                                                  [0]  [0]  [0]
                                                  [ ]  [ ]  [ ]
                                                  [0]  [0]  [0]
                                                  [ ]  [ ]  [ ]
                                                  [1]  [0]  [0]
                                                  [ ]  [ ]  [ ]
                                             V := {[1], [0], [0]}
                                                  [ ]  [ ]  [ ]
                                                  [1]  [0]  [0]
                                                  [ ]  [ ]  [ ]
```

```
                                                  [0]  [1]  [0]
                                                  [ ]  [ ]  [ ]
                                                  [0]  [0]  [1]

>  c:=1;bb:=1;
                                                     c := 1

                                                    bb := 1

>  for i from 1 to nops(a) do if (V[1])[i]=1 then
>  c:=c*a[i];bb:=bb*b[i]; print(c);fi ;od;
                                                      -115

                                                      2645

                                                     13225

>  c:=sqrt(c); igcd(bb-c,n);#pas de chance on a eu bb=c mod n
                                                    c := 115

                                                       1

#On essaye un n plus grand:
> n:=nextprime(1000)*nextprime(6000);
                                                  n := 6061063

>   x:=evalf(sqrt(n),100);
      x := 2461.922622667089429660188502227354815824452917551795623284829626538477368415657639553934527943022470

>   a:=[]:b:=[]:uu:=0:u:=1:vv:=1:v:=0:B:=floor(x):
>   for i from 1 to 10 do
>  oldu:=u:u:=B*u+uu:uu:=oldu:oldv:=v:v:=B*v+vv:vv:=oldv:
>  a:=[op(a),mods((u)^2,n)]: b:=[op(b),u]:
>  x:=evalf((1/(x-B),100)):B:=floor(x):
>  od: evalf(u/v,100);#on verifie que ca tend vers racine  de n
        2461.922622667103232143692408438187005940408812385986248187473689134197109312877122409841433182094579

>  P:={-1} union facteursliste({op(a)});P:=[op(P)];
                         P := {-1, 2, 3, 7, 17, 19, 41, 43, 71, 103, 109, 127, 283, 503, 757}

                         P := [-1, 2, 3, 7, 17, 19, 41, 43, 71, 103, 109, 127, 283, 503, 757]

>  A:=alpha(a,P);
                         [1    1    1    0    0    0    0    0    0    0    0    0    0    0    1]
                         [                                                                      ]
                         [0    0    1    0    0    0    0    0    0    0    0    1    0    0    0]
                         [                                                                      ]
                         [1    0    0    0    0    0    1    0    0    1    0    0    0    0    0]
                         [                                                                      ]
                         [0    1    3    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    1    0]
                    A := [                                                                      ]
                         [0    0    2    0    0    2    0    0    0    0    0    0    0    0    0]
                         [                                                                      ]
                         [1    1    0    0    1    0    0    1    0    0    0    0    0    0    0]
                         [                                                                      ]
                         [0    0    0    1    0    0    0    0    0    0    0    0    1    0    0]
                         [                                                                      ]
                         [1    1    1    0    0    0    0    0    0    0    1    0    0    0    0]
                         [                                                                      ]
                         [0    0    0    1    0    0    0    0    1    0    0    0    0    0    0]

> V:=(Nullspace(Transpose(A)) mod 2);
                                                       [0]
                                                       [ ]
                                                       [0]
                                                       [ ]
                                                       [0]
                                                       [ ]
                                                       [0]
                                                       [ ]
                                                       [0]
                                                  V := {[ ]}
                                                       [1]
                                                       [ ]
                                                       [0]
                                                       [ ]
                                                       [0]
                                                       [ ]
                                                       [0]
                                                       [ ]
                                                       [0]

>  c:=1;bb:=1; for i from 1 to nops(a) do if V[1][i]=1 then
                                                     c := 1

                                                    bb := 1

>  c:=c*a[i];bb:=bb*b[i]; fi ;od;c:=sqrt(c); igcd(bb-c,n);
                                                    c := 57

                                                      6007

>
> quit
bytes used=2950168, alloc=2358864, time=0.11
```