1 Calculer pour $nin at$: $u_nv_{n-1}-u_{n-1}v_n$. Montrer que pour tout $alpha$ on a $[a_0,a_1,...,a_{n-1},alpha]= rac{alpha u_{n-1}+u_{n-2}}{alpha v_{n-1}+v_{n-2}}$ En d'eduire lorsque les a_n sont dans $ at^*$ que rac{u_n}{v_n}$ est la fraction irr'eductible correspondant 'a $[a_0,a_1,dots,a_n]$.

2 On fait une proc'edure facteurs qui donne les facteurs d'un entier

3
```
facteurs:=proc(n)
l:=ifactors(n);
{seq(i[1],i=l[2])};
end proc;
```

```
// Warning: l,i, declared as global variable(s)
// End defining facteurs
```

```
proc(n)
  l:=maple_ifactors(n);
  {seq(i[1],i=(l[2]))};

end;
```

4 on retourne un ensemble plutot qu'une liste pour ne pas repeter de facteurs.

5
```
facteurs(123456789); {1,2} union {1,3};
```

```
( [3   3607   3803 ], [1   2   3 ] )
```

6
```
facteursliste:=proc(a)
l:={};
for i in a do l:=l union facteurs(i); od;
end proc;
```

```
// Warning: l,index,i,facteurs, declared as global variable(s)
// End defining facteursliste
```

```
proc(a)
  l:={ };
  for index from 1 to nops(a)+1 do
  i:=a[index];
  l:=l union (facteurs(i));
   od;;

end;
```

7
```
facteursliste([123456789,5*7,7*13]);#on teste
```

```
[3   3607   3803   5   7   13 ]
```

8 on utilise mods et non modp. recherche d'une assez bonne liste pour n

9
```
n:=nextprime(100)*nextprime(200);B:=floor(evalf(sqrt(n)))+1;
```

```
( 21311,  146 )
```

10 On fait une illustration a la main:

11
```
for i from 0 to 6 do print(ifactor(mods((B+i)^2,n))); od;
```

```
5
2*149
593
2*5*89
29*41
2*5*149
11*163
```

12 on voit une relation entre i=0,1,5

13
```
P:={-1,2,5,149};bb:=[B,B+1,B+5];
```

```
( [-1   2   5   149 ], [146   147   151 ] )
```

14
```
a:=[seq(mods(bb[i]^2,n),i=1..3)];
```

```
[5   298   1490 ]
```

15
```
c:=P[2]*P[3]*P[4];b:=bb[1]*bb[2]*bb[3];
```

```
( 1490,  3240762 )
```

16
```
igcd(c-b,n);#rate! on a b=c [n]
```

```
21311
```

17 on peut aussi chercher autour de sqrt(k*n)

18
```
B:=floor(evalf(sqrt(3*n)))+1;
```

```
253
```

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

```
2^2*19
11*53
2^2*3*7*13
7*229
2^2*23^2
3*877
2^2*787
```

```
1
```

20 i=4 est un carre, c'est particulierement favorable.

```
21 c:=sqrt(mods((B+4)^2,n));b:=B+4;igcd(c-b,n);#OK 211 divise n
```

```
( 46, 257, 211 )
```

22 un peu moins particulier:

```
23 B:=floor(evalf(sqrt(4*n)))+1;
```

```
292
```

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

$$2^3 \cdot 5 \cdot 89$$

25 le produit des 2 premiers est un carre.

```
26 c:=2*5*11;b:=B*(B+1);igcd(b-c,n);
```

```
( 110, 85556, 101 )
```

27 On automatise ce que l'on vient de faire.

```
28 a:={};b:={};for i from 1 to 10 do if max(op(facteurs(mods((B+i)^2,n))))<500
   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)];
```

```
{ }, { }, []293 294 295 298 300 301 302 [], []-1 5 11 2 149 13 137 89 29 41 487 [], [-1 5 11 2 149
```

```
29 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;
```

```
// Warning: nn,k, declared as global variable(s)
// End defining maxpow
```

```
proc(p,n)
  nn:=n;
  k:=0;
  if p=-1 then
    k:=(1-sign(n))/2
  fi ;
  while (irem(nn,p)=0) and (p>1) do
    nn:=nn/p;
    k:=k+1;
  od;;
  k;

end;
```

```
30 maxpow(5,5^3*7*2);
```

```
3
```

```
31 alpha:=proc(a,P)
   matrix([seq([seq(maxpow(pj,ai),pj=P)],ai=a)]);
   end proc;
```

```
// Warning: maxpow,pj,ai, declared as global variable(s)
// End defining alpha
```

```
proc(a,P)
  matrix(seq([seq(maxpow(pj,ai),pj=P)],ai=a));

end;
```

```
32 Nullspace(matrix([[2]])) mod 2;
```

```
[-1 ]
```

33 A:=alpha(a,P);V:=(Nullspace(transpose(A)) mod 2);

$$\left(\begin{bmatrix} 0&1&2&0&0&0&0&0&0&0&0\\ 0&0&0&3&1&0&0&0&0&0&0\\ 0&0&0&0&0&1&1&0&0&0&0\\ 0&1&0&3&0&0&0&1&0&0&0\\ 0&0&0&2&0&0&0&0&1&1&0\\ 0&0&1&0&0&0&0&0&0&0&1\\ 0&1&0&3&1&0&0&0&0&0&0 \end{bmatrix}, [1\ 1\ 0\ 0\ 0\ 0\ 1]\right)$$

34 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]; fi ;od;c:=sqrt(c); igcd(bb-c,n);

(1, 1, 26014884, 65560, 101)

35 on y incorpore des fractions continues. Attention a la precision: (pour xcas
On utilise pari ou with(Numtheory) et cfrac pour maple

36 pari()

All PARI functions are now defined with the pari_ prefix.
PARI functions are also defined without prefix except:
abs acos acosh arg asin asinh atan atanh binomial bitand bitor bitxor ceil charpoly concat conj content cos cosh divisors erfc eval exp fa
Note that p-adic numbers must have O argument quoted e.g. 905/7+O('7^3')
Type ?pari for short help
Inside xcas, try Help->Manuals->PARI for HTML help

37 contfrac(sqrt(n),7);

[145 1 57 2 1 1 11]

38 contfrac(sqrt(n),100);

[145 1 57 2 1 1 11 12 1 1 1 1 4 2 3 9 7 1 3 1 1 1 1 2 22 13]

39 il faut augmenter la precision de sqrt(n) pour avoir suffisamment de chiffres

40 dim(contfrac(sqrt(n),100));

26

41 contfrac(evalf(sqrt(n),106),100);

[145 1 57 2 1 1 11 12 1 1 1 1 4 2 3 9 7 1 3 1 1 1 1 2 22 13 4 2 2 2 4 13 22 2 1

42 x:=evalf(sqrt(n),10):B:=floor(x):a:=[B]:

(Done, Done, Done)

43 Si on ne travaillequ'avec 10 chiffres, on obtientun erreur rapidement:

44 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.

(Done, [145 1 57 2 1 1 11 12])

45 On travaille avec 100 chiffres.

46 x:=evalf(sqrt(n),100);

0.14598287570807748682380132824925340174649633185930741965792268383222904910364987590654397104431881760e3

47 a:={}:b:={}:uu:=0:u:=1:vv:=1:v:=0:B:=floor(x):

(Done, Done, Done, Done, Done, Done, Done)

48 On remarque que u*vv-v*uu vaut +/- 1

```
49 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

Done, 0.14598287570121051077649837614408030705639208739297313256569235311485090050191910245054620608207853535
```

```
50 P:={-1} union facteursliste(a):P:=[op(P)];a:=[op(a)];b:=[op(b)];

Done, [-1  2  11  13  5  23  83 ], [-286  5  -115  121  -166  25  -23 ], [145  146  8467  17080  25547  42627
```

```
51 A:=alpha(a,P);

[1  1  1  1  0  0  0]
[0  0  0  0  1  0  0]
[1  0  0  0  1  1  0]
[0  0  2  0  0  0  0]
[1  1  0  0  0  0  1]
[0  0  0  0  2  0  0]
[1  0  0  0  0  1  0]
```

```
52 V:=(Nullspace(transpose(A)) mod 2);

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

```
53 c:=1;bb:=1;

( 1,  1 )
```

```
54 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;
   c:=sqrt(c); igcd(bb-c,n);#pas de chance on a eu bb=c mod n

c:121
( undef,  11,  101 )
```

```
55

undef
```

```
56 On essaye un n plus grand:
```

```
57 n:=nextprime(1000)*nextprime(6000);

6061063
```

```
58  x:=evalf(sqrt(n),100);

0.24619226226670894296601885022273548158244529175517956232848296265384773684156576395539345279430224694e4
```

```
59  a:=[]:b:=[]:uu:=0:u:=1:vv:=1:v:=0:B:=floor(x):

( Done,  Done,  Done,  Done,  Done,  Done,  Done )
```

```
60 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:

Done
```

```
61 evalf(sqrt(n)-u/v,100);#on verifie que ca tend vers racine  de n

0.13802483503006210832190115055804834190624902644062505710740807219482855906905239072109545047582038290e-10
```

```
62 plotlist(ll);

Bad Argument Value
```

```
63 P:={-1} union facteursliste({op(a)});P:=[op(P)];
```

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

```
64 A:=alpha(a,P);
```

1	1	1	1	0	0	0	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0	0	0	0	0	0	0
1	0	0	0	0	1	1	0	0	0	0	0	0	0
0	1	3	0	0	0	0	1	0	0	0	0	0	0
1	0	0	0	0	0	0	0	1	0	0	0	0	0
0	0	2	0	0	0	0	0	0	2	0	0	0	0
1	1	0	0	0	0	0	0	0	0	1	1	0	0
0	0	0	0	0	0	0	1	0	0	0	0	1	0
1	1	1	0	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	1	0	0	0	0	0	1

```
65 V:=(Nullspace(transpose(A)) mod 2);
```

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

```
66 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]; fi ;od;c:=sqrt(c); igcd(bb-c,n);
```

```
( 1,  1,  undef,  57,  6007 )
```

```
67 On illustre la convergence de u_k/v_k vers sqrt(n), d'abord en montrant que la
   difference est proche de 0, puis en illustrant le fait que
   (u_k^2-n.v_k^2)/sqrt(n) a l'air borne.
```

```
68 x:=evalf(sqrt(n),100);
    a:=[]:b:=[]:uu:=0:u:=1:vv:=1:v:=0:B:=floor(x):ll:=[];
   for i from 1 to 50 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):
   ll:=augment(ll,evalf((u*u-n*v*v)/sqrt(n),10));
   od:
```

```
0.24619226226670894296601885022273548158244529175517956232848296265384773684156576395539345279430224694e4
```

```
69 evalf(sqrt(n)-u/v,100);#on verifie que ca tend vers racine  de n
```

```
-0.5102790448492065411477188030387098164351381448901018788229213368022401769549928372292685611431088225 3e-53
```

```
70 plotlist(ll);
```