```
1 restart;maple_mode(1);cas_setup(0,0,0,1,0,1e-10,10,[1,50,0,25],0,0,0); #radians,pas de cmplx, pas de  Sqrt
```

Menu

```
2 l:=[];
```

[]

Menu

```
3 for n from 10^4 to 10^5 by 1000  do l:=[op(l),[n,time(27^(2^n) mod 107)]] ; od;
```

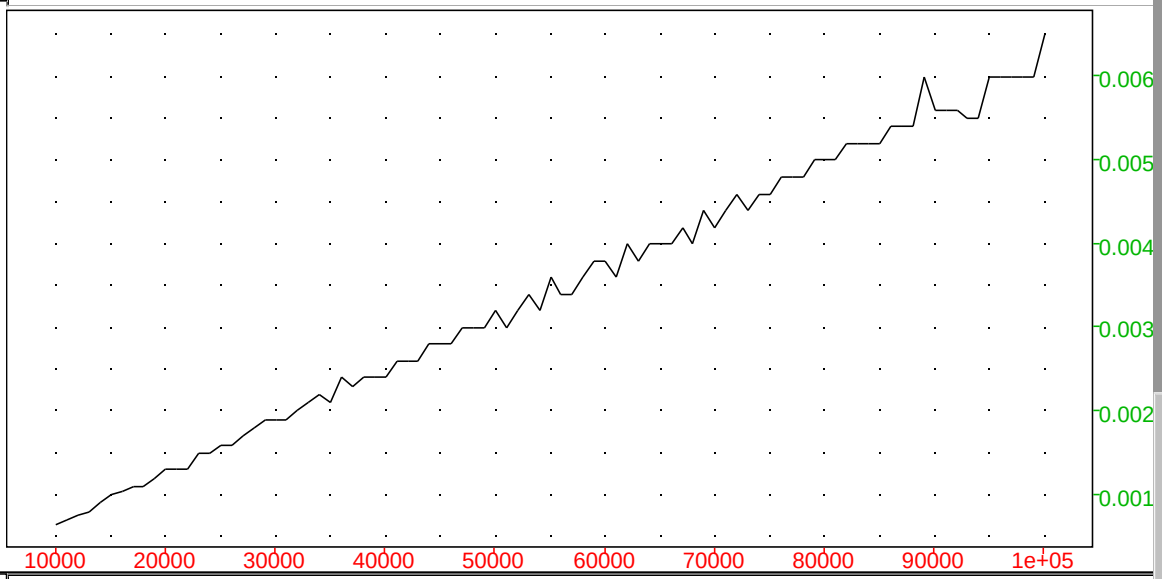
Evaluation time: 16.89

31000	0.0019
32000	0.002
33000	0.0021
34000	0.0022
35000	0.0021
36000	0.0024
37000	0.0023
38000	0.0024
39000	0.0024
40000	0.0024
41000	0.0026
42000	0.0026
43000	0.0026
44000	0.0028
45000	0.0028
46000	0.0028
47000	0.003
48000	0.003
49000	0.003
50000	0.0032
51000	0.003
52000	0.0032
53000	0.0034
54000	0.0032

Menu

```
4 Le coup semble lineaire (en n) sur ce dessin, ce qui illustre bien que le
  calcul de 27^A est en O(ln(A))
```

5 plotlist(l);



6 ----------EXERCICE----------

7

8 Prog Edit Add | nxt | OK | Save

```
puis:=proc(a,n)
local A,B,C;
A:=1;B:=a;C:=n;
while C>0 do
if irem(C,2)=1 then A:=A*B;C:=(C-1)/2;B:=B*B;
else C:=(C)/2;B:=B*B;
fi;
od;
A;
end;
```

// Success
// End defining puis

```
proc(a,n)
 local A,B,C;
 A:=1;
 B:=a;
 C:=n;
 while C>0 do
  if irem(C,2)=1 then
   A:=A*B;
   C:=(C-1)/2;
   B:=B*B else
   C:=C/2;
   B:=B*B
  fi ;
 od;;
 A;

end;
```

Menu

9 puis(2,7);

128

Menu

10 debug(puis(2,71));)

Evaluation time: 4.55

236118324143482260684 8

Menu

11 convert(71,base,2);

[1 1 1 0 0 0 1]

Menu

12 ----------

```
13 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 divis
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
```

Menu

```
14 p:=nextprime(2^256);
```

```
115792089237316195423570985008687907853269984665640564039457584007913129640233
```

Menu

```
15 znprimroot(p);
```

```
Evaluation time: 36.82
5 % 115792089237316195423570985008687907853269984665640564039457584007913129640233
```

Menu

```
16 p:=nextprime(2^256+987654327);
```

```
115792089237316195423570985008687907853269984665640564039457584007914117294279
```

Menu

```
17 znprimroot(p);
```

```
Evaluation time: 6.39
11 % 115792089237316195423570985008687907853269984665640564039457584007914117294279
```

Menu

```
18 ----------EXERCICE----------
```

```
19 b,e pas trop petits par rapport a p, pour que le modulo p melange
vraiment.
```

```
20 message:="ABCDA, Remarquez que des lettres identiques ne sont pas codees de la meme maniere!";asc(message
```

```
ABCDA, Remarquez que des lettres identiques ne sont pas codees de la meme maniere! , [65 66 67 68
```

Menu

```
21 asc(messsage)-[seq(65,i=1..length(message))];
```

```
[-65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -65 -6
```

Menu

```
22 b:=12345;e:=54321; igcd(e,p-1); B:=powmod(b,e,p);
```

Menu

```
23 l:=asc(message);k:=[seq(rand(50),i=1..length(message))];
```

```
[65 66 67 68 65 44 32 82 101 109 97 114 113 117 101 122 32 113 117 101 32
```

Menu

```
24 C:=[seq([(b^k[i]) mod p,(l[i]*B^k[i]) mod p],i=1..length(message))];#le message crypte
```

```
25530126296115036384884547520768961631647262347515344902161510733840097609620 323338779
1546622149409141311021651977071012958492308459472656251 5154103351
25530126296115036384884547520768961631647262347515344902161510733840097609620 637402894
8383894234713303303962791290357317080872540169843839132802563720715775438534 8 555441424
1881365963625 254272009
19093050434455849484562293656941654972587547932189941406251 3347443001
9331252831503415355673333857300789057639494457639496873172071109417596153611 5 542035119
5589677923864347522303028742474553381430778037551229178935875716815541662071 1 629433355
10148504227039125158587149603293150717285156251 688908941
9913431080275642386308893525303407393782540486895806581976392372169183302574 1 119905440
64757108331831581111651418831446612441551761322012119584329646413730195378941 720706072
14610380094935155905430231855773263044723444926954568263613653145088785857901 178483464
1461038009493515590543023185577326304472344492695456826361365314508878585790 8541873751
2816497979126341463794170348115334189551560900989497879177343365891315080452 8 149141297
7403570333251010134855916465056259190334779940442187406276351283890348761760 6 302228851
8426441284416026865313712322769923446065070859887789592205552877928713433850 9 844892534
8175694242935189656968723135457086296646537419455435654307842785235488023567 424876893
```

1	100
87983420589090484253501881241718375425298023768817821469016954192383010151312	72545935[illegible]
12345	45178615[illegible]
12345	60274767[illegible]
23797289700470669717102937621540360968594681687063341376855421797426835601923	48456203[illegible]
28164979791263414637941703481153341895515609009894978791773433658913150804528	10267033[illegible]

Menu

25 ee:=p-1-e;#c'est -e [p-1] qui nous interesse

115792089237316195423570985008687907853269984665640564039457584007914117239957

Menu

26 L'astuce est que (B^k)=((b^e)^k) qui vaut par commutativit'e des la composition des application B:x->x^k et A:x->x^e ((b^k)^e).
Le principe a retenir, est qu'en crypto on a juste besoin d'operations A,B qui commutent, et que la donnee de A de donne pas A^(-1)

27 decrypt:=[seq(irem(powmod(C[i][1],ee,p)*C[i][2],p),i=1..length(message))];

[65 66 67 68 65 44 32 82 101 109 97 114 113 117 101 122 32 113 117 101 32

Menu

28 char(decrypt);

ABCDA, Remarquez que des lettres identiques ne sont pas codees de la meme maniere!

Menu

29