

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  
Warning: some commands like subs might change arguments order

2 l:={1,4,5,6,7};a:=rand(10):  
( [1 4 5 6 7 ], Done )

3 member(a,l);  
0

4 a; #verification.  
9

5 7 + 4 mod 3; (7+4)mod 3;  
( 8, 2 )

6 maple\_mode(1); # en mode maple  
Warning: some commands like subs might change arguments order

7 ln(exp(1));ln(e); #ca n'est que la lettre e, pas le reel.  
( 1, ln( e ) )

8 maple\_mode(0);  
Warning: some commands like subs might change arguments order

9 evalf(e); //la doc fait reference au mode xcas.  
2.718281828

10 maple\_mode(1);  
Warning: some commands like subs might change arguments order

11 evalf(exp(1));  
2.718281828

12 ln(exp(1));  
1

13 log(exp(1)); # les 2 marchent  
1

14 bete comme le crible d'erastotene sont en  $O(p)$ , et bete n'use pas de memoire.  
donc asymptotiquement, le crible n'est pas avantageux.

15 Prog Edit Ajouter     

```
bete:=proc(n)
i:=3;
a:=0;
SQ:=evalf(sqrt(n)); #pour ne le faire qu'une fois. ne pas le mettre dans le while!
while (i<SQ)
do if (n mod i) <>0 then i:=i+2;
else a:=i;i:=n;
fi ;
od;
if a<>0 then a; else n fi;
end;
```

// Warning: i a SQ declared as global variable(s)  
// End defining bete

```
proc(n)
i:=3;
a:=0;
SQ:=evalf(sqrt(n));
while i<SQ do
if (irem(n,i)<>0 then
i:=i+2 else
a:=i;
i:=n;
fi ;
od;;
if a<>0 then
a else
n
fi ;
end;
```

16 i:=4;N:=nextprime(10<sup>i</sup>+5432)\*nextprime(2\*10<sup>i</sup>+1234);  
( 4, 328032433 )

17 bete(N);



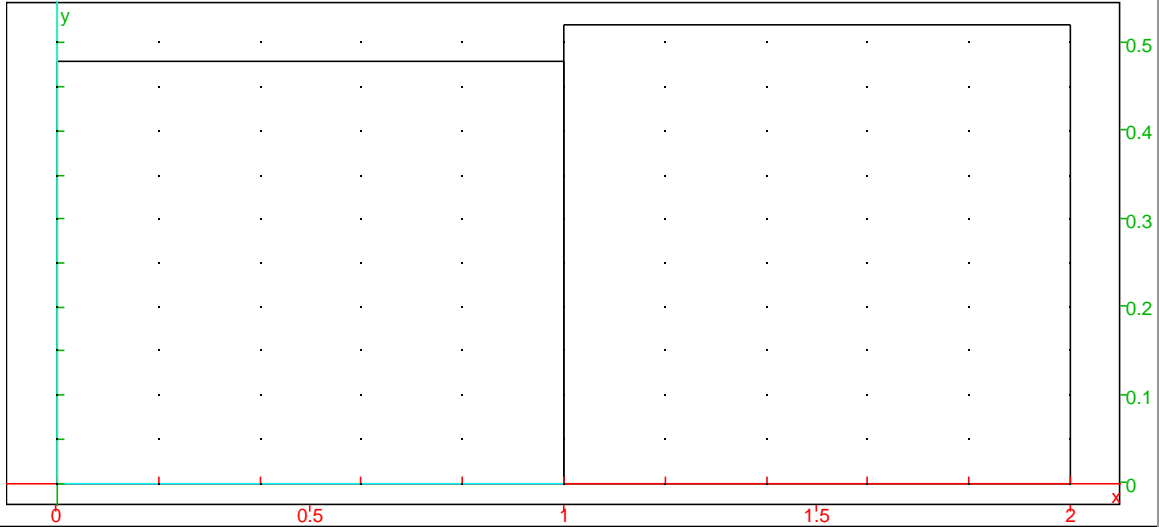
```
29 l:=[];
for i from 7 to 30 do
N:=nextprime((rand(3^(i))))*nextprime((rand(2^(i+1))));
t:=comptep(N);
print(i,t);
l:=append(l,t);
N:=nextprime((rand(3^(i))))*nextprime((rand(2^(i+1))));
t:=comptep(N);
print(i,t);
l:=append(l,t);
N:=nextprime((rand(3^(i))))*nextprime((rand(2^(i+1))));
t:=comptep(N);
print(i,t);
l:=append(l,t);
od;
```

```
8,1.701143931
9,1.455468278
9,0.9145661413
9,0.5862383625
10,0.3719760372
10,0.7298051654
10,0.7758601925
11,0.6510247955
11,2.540811749
11,2.088130666
12,1.251086484
12,0.9145661413
12,0.3170975517
13,0.7556357015
13,0.7245847576
13,1.104268102
14,1.227507969
14,1.500380418
14,0.7074527316
15,1.971208849
15,0.9738950371
15,1.014818273
16,0.8409397775
16,0.5392030995
16,2.36640645
17,0.5517617252
17,0.3423219968
17,1.672435481
18,0.1466897604
18,1.03296731
18,1.040938929
19,0.5106115043
19,2.310941976
19,1.710424508
20,0.9315111102
20,0.176860132
20,1.699539932
21,1.020533007
21,1.380618293
21,1.56378505
22,0.6502084347
22,1.554181147
22,2.459804425
23,0.80462712
23,0.3011148273
23,2.225372059
24,2.328030734
24,1.691982944
24,2.225962339
25,1.231496278
25,0.8828343427
25,0.9717050815
26,1.744090422
26,1.067180196
26,0.5036355722
27,0.9768428688
27,2.40148644
27,0.9365314714
28,1.174135102
28,0.4215494353
28,1.392917127
29,1.026060136
29,0.8538001091
29,0.758136072
30,1.292273334
30,0.8924453608
```





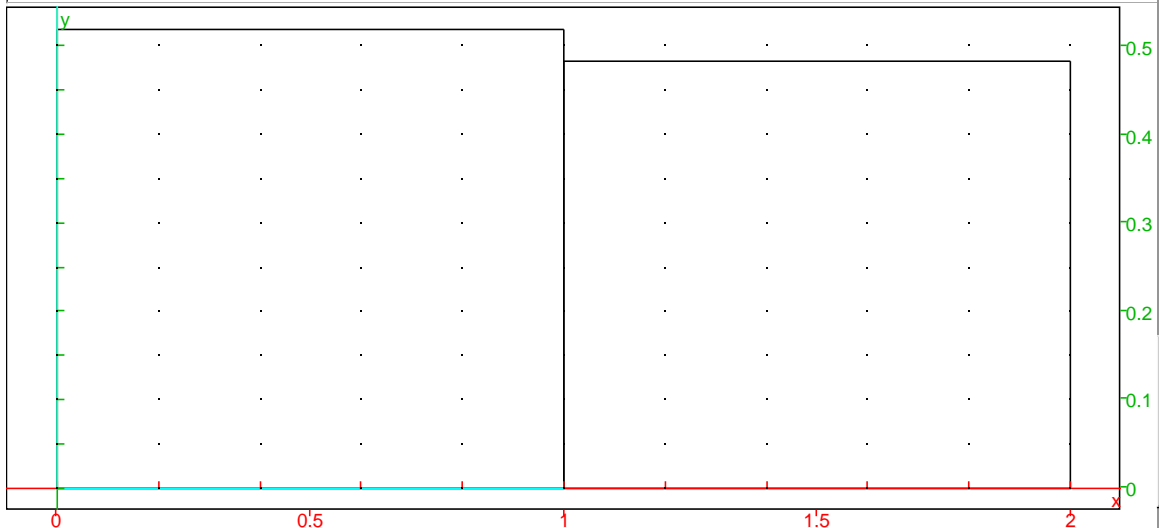
```
48 histogram(classes(l,0,1)); #On commence a 0, largeur constante 1.
```



```
49 l:=seq(rand(2),i=1..1000);
```

Done

```
50 histogram(classes(l,0,1));
```



```
51 N:=nextprime(10^6)*nextprime(2*10^6);
```

2000009000009

```
52 u:=n->if n=0 then 2 else ((u(n-1))^2+1) mod N fi;
```

```
// Warning: u N declared as global variable(s)  
// End defining u  
u: recursive definition
```

```
if n=0 then  
2 else  
irem((u(n-1))^2+1,N)  
n -> fi
```

```
53 u(10); # u(10000);Error, (in u) too many levels of recursion
```

1997309223146

```

55 etudesuite := proc(n,M)
    local x,i;
    x:=12345;l:=[];
    for i from 1 to M do
        x:=(x^2+1) mod n ;
        l:=[op(l),x];
        od ;
    l;
end;

```

// Warning: i declared as global variable(s)  
// End defining etudesuite

```

proc(n,M)
local x,i;
x:=12345;
l:=[];
for i from 1 to M+1/2 do
x:=irem(x^2+1,n);
l:=[op(l),x];
od;;
l;
end;

```

```

56 donnees:=(etudesuite(N,2000));

```

Evaluation time: 2.99

152399026 1358617644169 1831711515025 190274858284 1218880810174 893513062655 523067646767 1742351

```

57 cldonnees:=classes(donnees,0,N/40); #40 classes

```

0.0 .. 5.0000225e+10	58
5.0000225e+10 .. 1.0000045e+11	50
1.0000045e+11 .. 1.50000675e+11	54
1.50000675e+11 .. 2.000009e+11	46
2.000009e+11 .. 2.50001125e+11	44
2.50001125e+11 .. 3.0000135e+11	48
3.0000135e+11 .. 3.50001575e+11	44
3.50001575e+11 .. 4.000018e+11	50
4.000018e+11 .. 4.50002025e+11	50
4.50002025e+11 .. 5.0000225e+11	48
5.0000225e+11 .. 5.50002475e+11	51
5.50002475e+11 .. 6.000027e+11	51
6.000027e+11 .. 6.50002925e+11	53
6.50002925e+11 .. 7.0000315e+11	48
7.0000315e+11 .. 7.50003375e+11	52
7.50003375e+11 .. 8.000036e+11	45
8.000036e+11 .. 8.50003825e+11	40
8.50003825e+11 .. 9.0000405e+11	50
9.0000405e+11 .. 9.50004275e+11	46
9.50004275e+11 .. 1.0000045e+12	57

