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 cube might change arguments order

Menu

2 time(p:=nextprime(10^50));

0.0025

Menu

3 time(isprime(p));

0.078125

Menu

4 time(is_pseudoprime(p));

0.00140625

Menu

5 le temps de isprime est trop long pour avoir ete utilise pour
trouver le nombre premier avec nextprime. La doc confirme bien que
nextprime donne un pseudopremier.

6 i:=20;n:=nextprime(10^i)*nextprime(20^i);

(20, 104857600000000000040894555000000000000000003549)

Menu

7 ifactor(n);

Evaluation time: 2.07812

$$10000000000000000039 \cdot 104857600000000000000000091$$

Menu

8 a partir de i=30 ca ne repond plus?

9 p:=nextprime(=nextprime(10^30)*nextprime(20^30));

1073741824000000000000000000061203284109000000000000000000000000008409

Menu

10 ifactor(p-1); #ca marche souvent

$$2^3 \cdot 3 \cdot 389 \cdot 733 \cdot 1569043746222576048238799828476023929007518023499814708952772 41$$

Menu

11 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

12 pari_isprime(p,1);

Evaluation time: 12.0625

2	7	1
3	2	1
389	2	1
733	2	1
1569043746222576048238799828476023929007518023499814708952772 41	2	2 3 5 7 56467 6553084925887974620 8

Menu

13 Les grands nombres premiers apparaissant dans le certificat sont eux
m{^e}me certifi{'e}s, ce qui explique les crochets emboit{'e}s.

14 p:=nextprime(10^13);i:=0;l:=[];

```
15 while i<10 do
   if ispseudoprime(4*p+1)>0 then l:=[op(l),4*p+1];i++; fi;
   p:=nextprime(p);
   od:l;
```

(Done, [40000000000733 40000000002629 40000000011053 40000000016309 40000000017917 400(Menu

```
16 p:= extprime(10^13+100);
```

10000000000129 Menu

```
17 g:=pari_znprimroot(p);
```

7 % 10000000000129 Menu

```
18 pari_znlog(3,Mod(g,p));
```

2552475531946 Menu

```
19 ifactor(p-1);
```

$2^7 \cdot 3 \cdot 98867 \cdot 263401$ Menu

```
20 p-1 a de petits facteurs premiers, on peut donc resoudre le probleme
   du log discret. Dans le cas suivant ca ne marche plus.
```

```
21 p:=l[3];
```

40000000011053 Menu

```
22 g:=pari_znprimroot(p);
```

2 % 40000000011053 Menu

```
23 pari_znlog(3,Mod(g,p)); #ca depasse la memoire de pari
```

Error in PARI subsystem

undef Menu

```
24 ----------------------------------EXERCICE-------------------------------------------
```

```
25 p:=l[3];
```

40000000011053 Menu

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

```
27 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

```
28 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 -65 -65 -65 Menu

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

(12345, 54321, 1, 33473253573817) Menu

```
30 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

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

380901004439[illegible]	1871610613957[illegible]
16088998776349	38338962478444
16140057393540	28774154633666
33604145840536	38170669619125
16140057393540	21090277825400
25616939055286	28970370733378
152399025	35603652517936
26072565355155	24716418195292
28871295273866	36886580726949
12345	2897413490631
16140057393540	29602536281495
18072155013312	11942674603400
9285222567600	3123239293725
8898364078510	30262442773278
2504656109032	5133093648656
18072155013312	21329380798072
10793236508135	35297819144575
8898364078510	39027147015182
22289850177970	22249200168147
2916562363890	4205025507103
6064246205996	10679585368370
10304518854412	19475890601838

Menu

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

39999999956731

Menu

```
33 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)
```

```
34 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

```
35 char(decrypt);
```

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

Menu

```
36 ----------Exercice----------
```

```
37 8*2+7 mod 7; (8*2+7) mod 7; #Attention mod est prioritaire sous xcas
```

(16, 2)

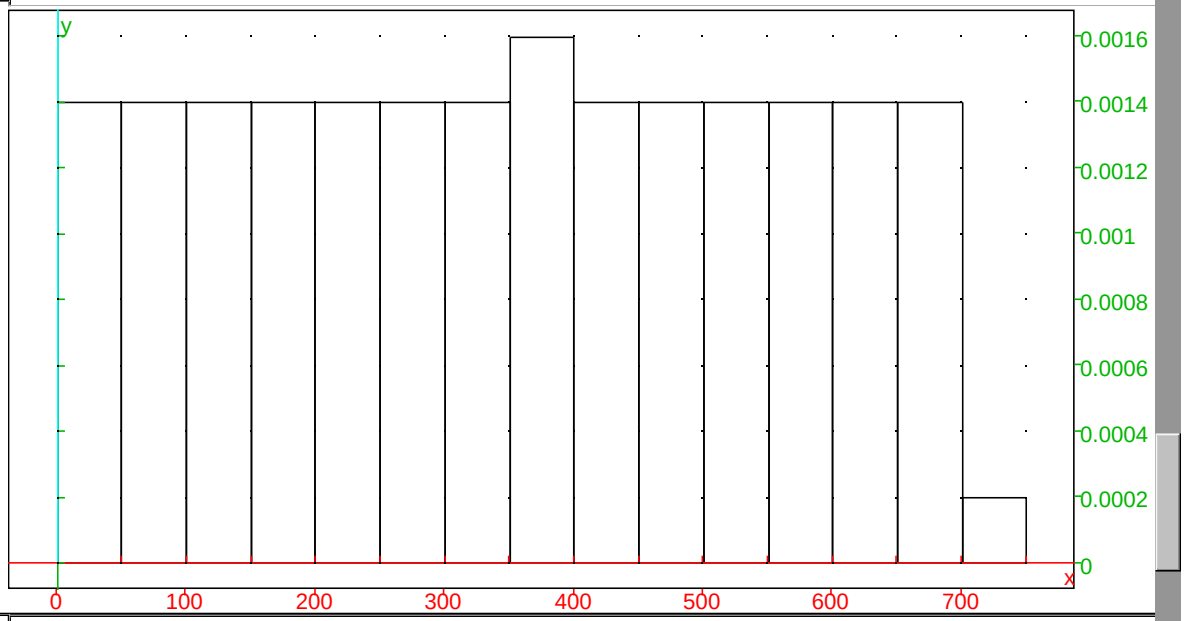
Menu

```
38 Si a=1, on a x_n=x_0+n.c [m], on fait donc par exemple: c=7,x0=1
```

```
39 l:=[seq((n*7+1) mod 1001,n=1..100)];
```

[8 15 22 29 36 43 50 57 64 71 78 85 92 99 106 113 120 127 134 141 148 155

```
40 histogram(classes(l,0,1001/20)); #C'est tout a fait plat. Trop!
```




```
41 Remplacer les crochets par des incolades cree un ensemble, ce qui simplifie
   automatiquement les elements egaux
```

```
42 nops(l);nops({op(l)}); #ils sont tous distincts
```

(100, 100)

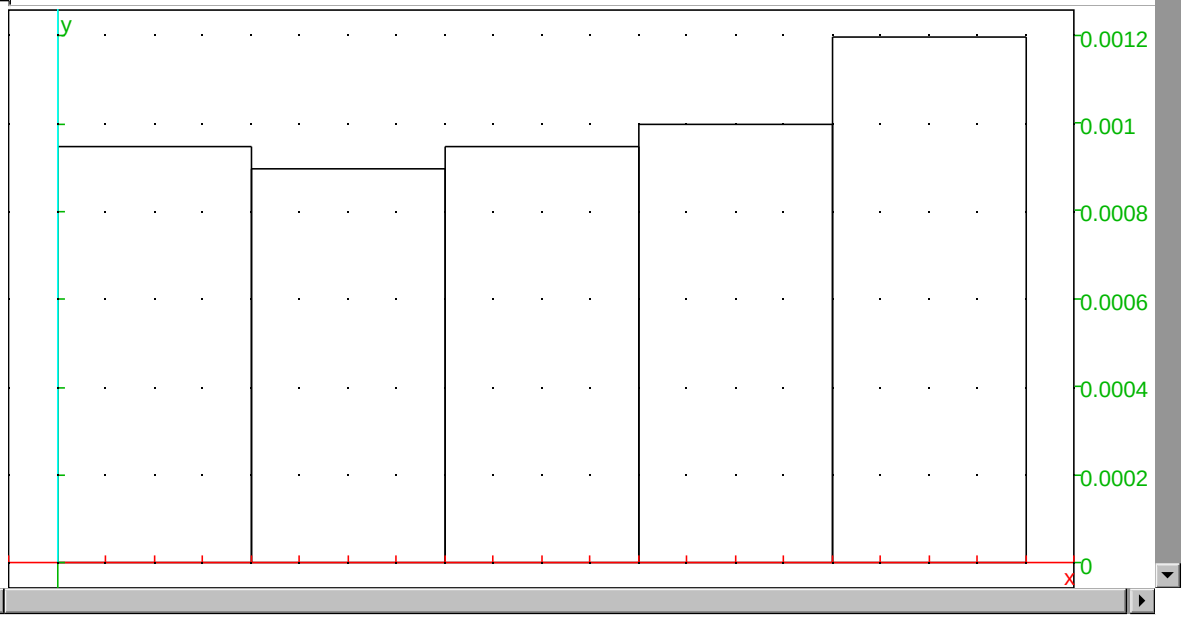
Menu

```
43 l:=[seq(rand(1001),n=1..100)];
```

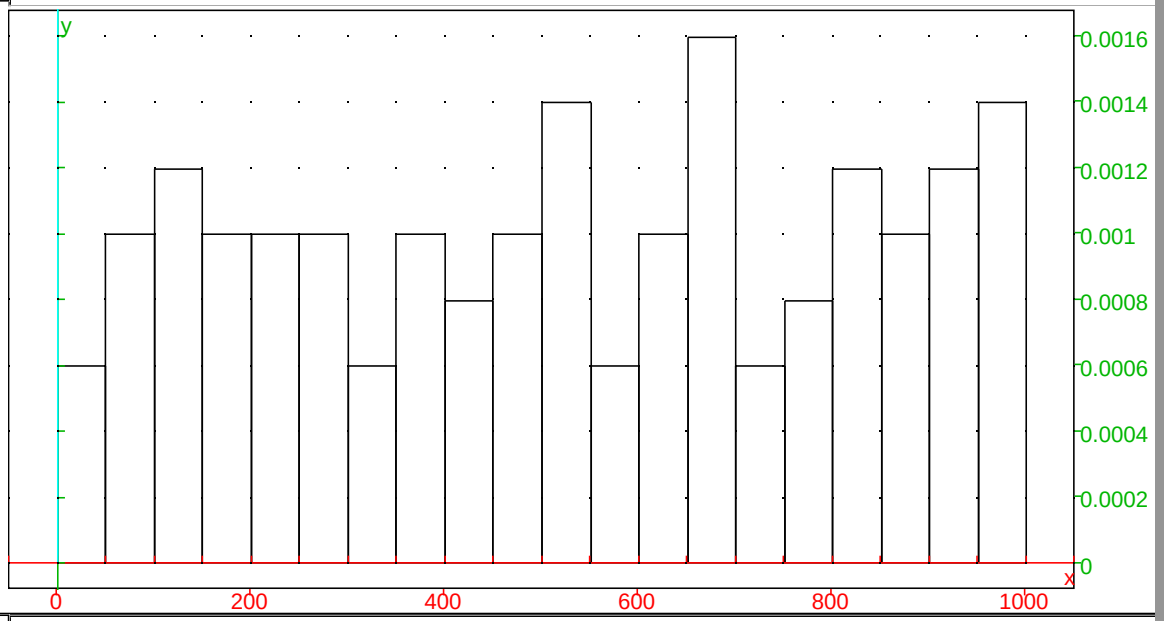
979 626 186 542 69 189 71 460 540 574 289 231 109 345 255 379 102 373 937

Menu

```
44 histogram(classes(l,0,1001/5));
```



45 histogram(classes(l,0,1001/20));



46 nops(l);nops({op(l)});#il y a bien des anniversaires identiques

(100, 96)

47 On a en fait 1001^100 suites possibles, et 1001!/901! suites dont tous les termes sont distincts.

48 1001!/901!/1001.^100;

0.00599058972

49 x:=1; a:=237;c:=54321;m:=10^4; l:=[];

(1, 237, 54321, 10000, [])

50 for i from 1 to 500 do x:=(a*x+c) mod m; l:=[op(l),x] ; od:

Done

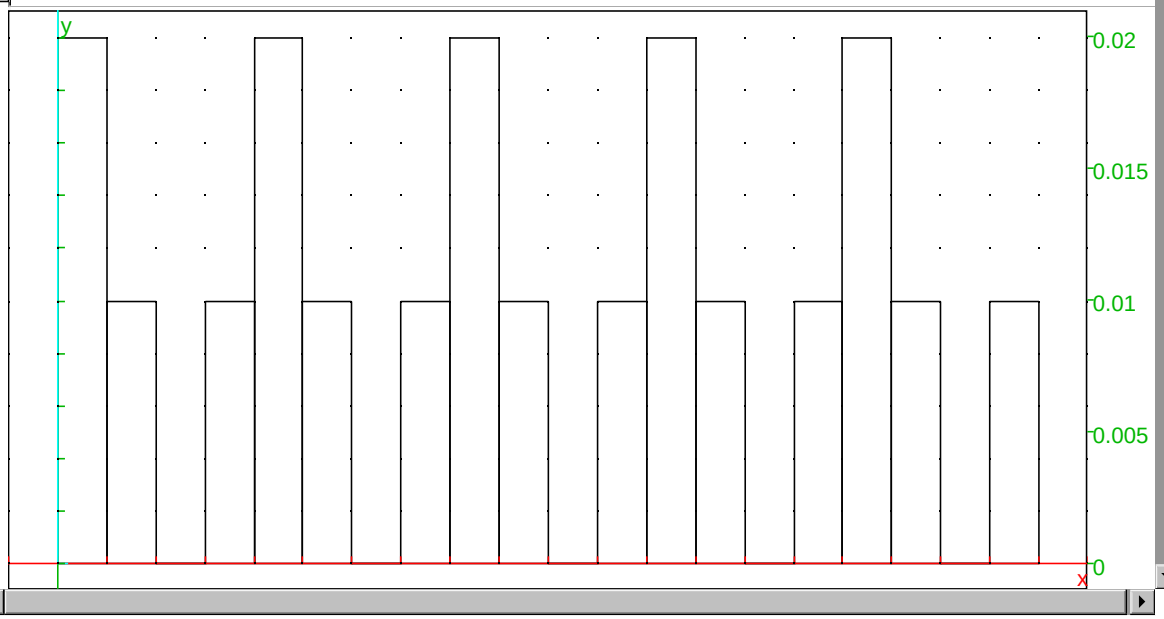
51 l1:=[seq(i mod 10^2,i=l)];

[58 67 0 21 98 47 60 41 38 27 20 61 78 7 80 81 18 87 40 1 58 67 0 21 98

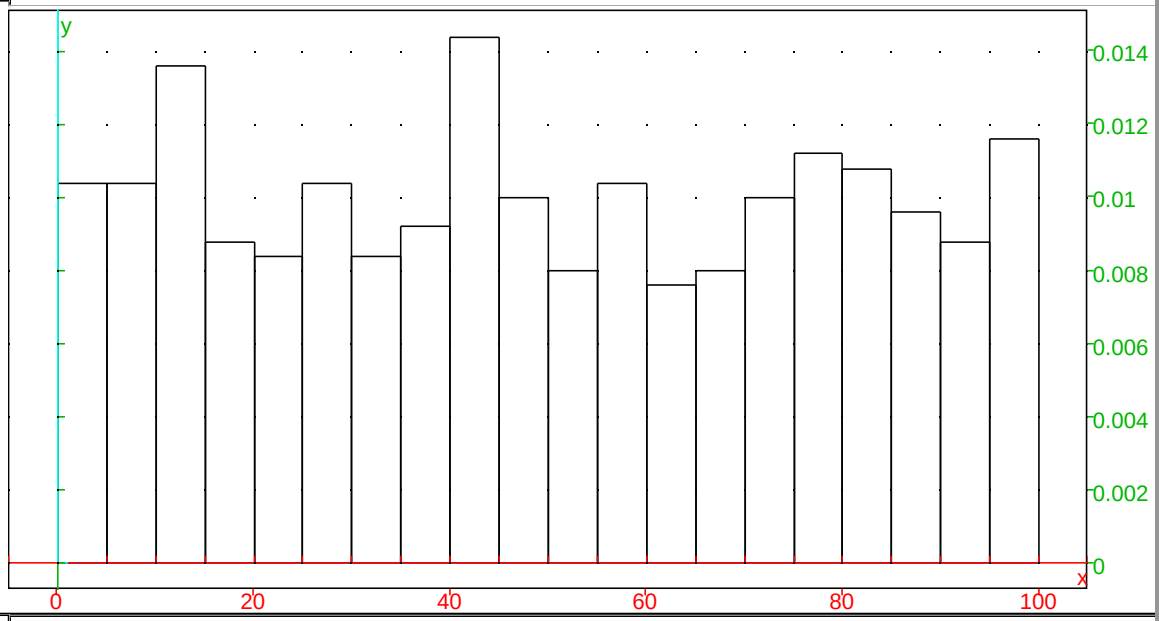
52 l2:=[seq(trunc(i/10^2),i=l)];

[45 45 67 22 6 97 43 76 52 57 16 82 21 5 44 60 55 20 89 31 92 84 10 13 73

53 histogram(classes(l1,0,5));



54 histogram(classes(l2,0,5));

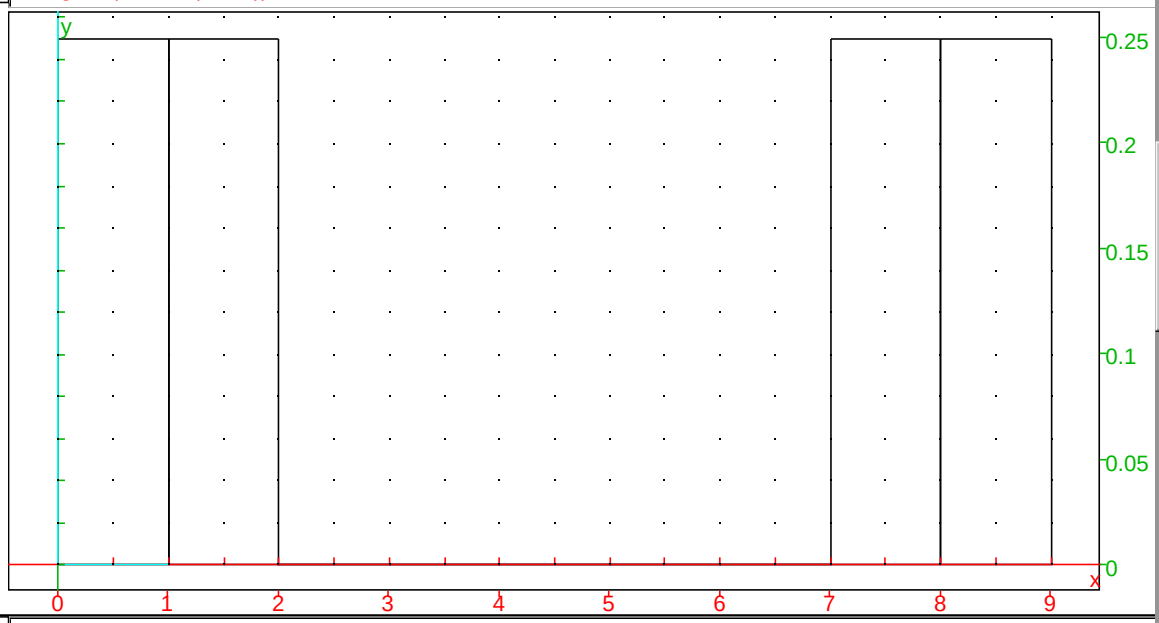


55 l1:=[seq(i mod 10,i=l)];

[8 7 0 1 8 7 0 1 8 7 0 1 8 7 0 1 8 7 0 1 8 7 0 1 8 7 0 1 8 7 0 1 8

Menu

56 histogram(classes(l1,0,1));



57 Si d est un diviseur de m et que $y_n=x_n [d]$, alors $y_{n+1}=ay_n+c [d]$
Donc si d=10^2, les 2 derniers chiffres de x_n ont p{'e}riode d'au plus d.

58 x:=1; a:=237;c:=54321;m:=10^4-1; l:=[];

(1, 237, 54321, 9999, [])

Menu

59 for i from 1 to 500 do x:=(a*x+c) mod m; l:=[op(l),x] ; od:

Done

Menu

60 l1:=[seq(i mod 10^2,i=l)];

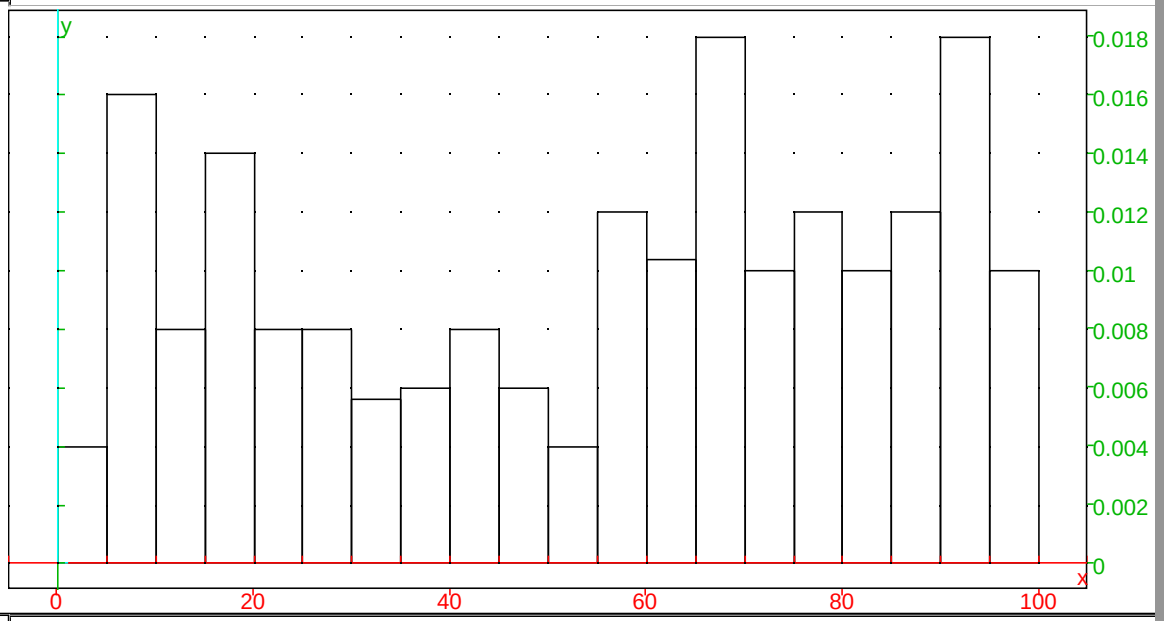
[63 65 70 22 31 78 27 68 10 93 68 17 72 91 56 79 62 81 60 61 25 92 6 69

Menu

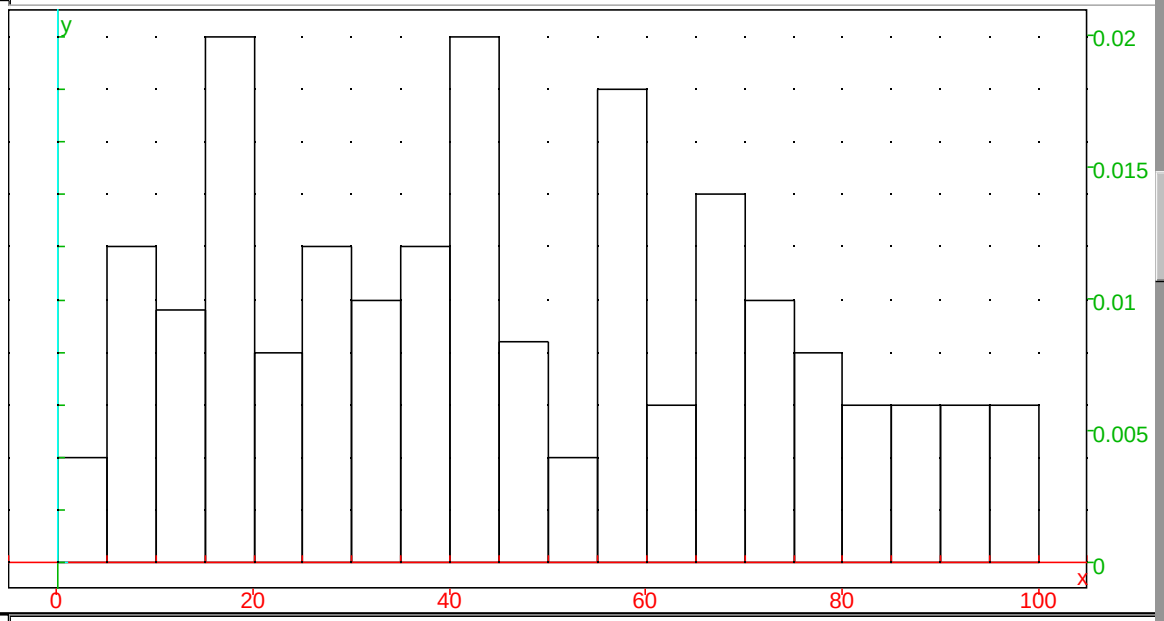
61 l2:=[seq(trunc(i/10^2),i=l)];

[45 58 44 38 2 90 60 28 41 84 73 7 42 68 76 89 25 15 90 17 17 31 9 90 38

62 histogram(classes(l1,0,5));



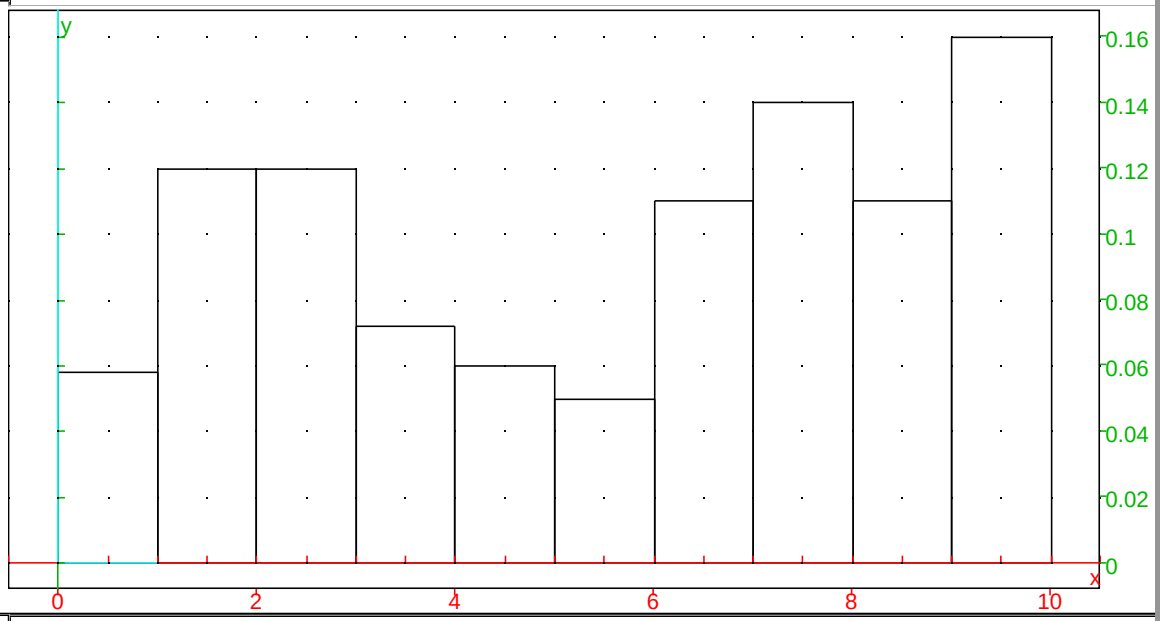
63 histogram(classes(l2,0,5));



64 l1:=[seq(i mod 10,i=l)];

[3 5 0 2 1 8 7 8 0 3 8 7 2 1 6 9 2 1 0 1 5 2 6 9 4 6 1 9 9 5 4 1 8

```
65 histogram(classes(l1,0,1));
```




```
66 x:=1; a:=237;c:=54321;m:=prevprime(10^4); l:=[];
```

(1, 237, 54321, 9973, [])

Menu

```
67 for i from 1 to 500  do x:=(a*x+c) mod m; l:=[op(l),x] ; od:
```

Done

Menu

```
68 l1:=[seq(i mod 10^2,i=l)];
```

[93 94 44 95 62 27 20 98 29 1 58 77 43 76 26 47 76 89 96 18 57 31 49 96

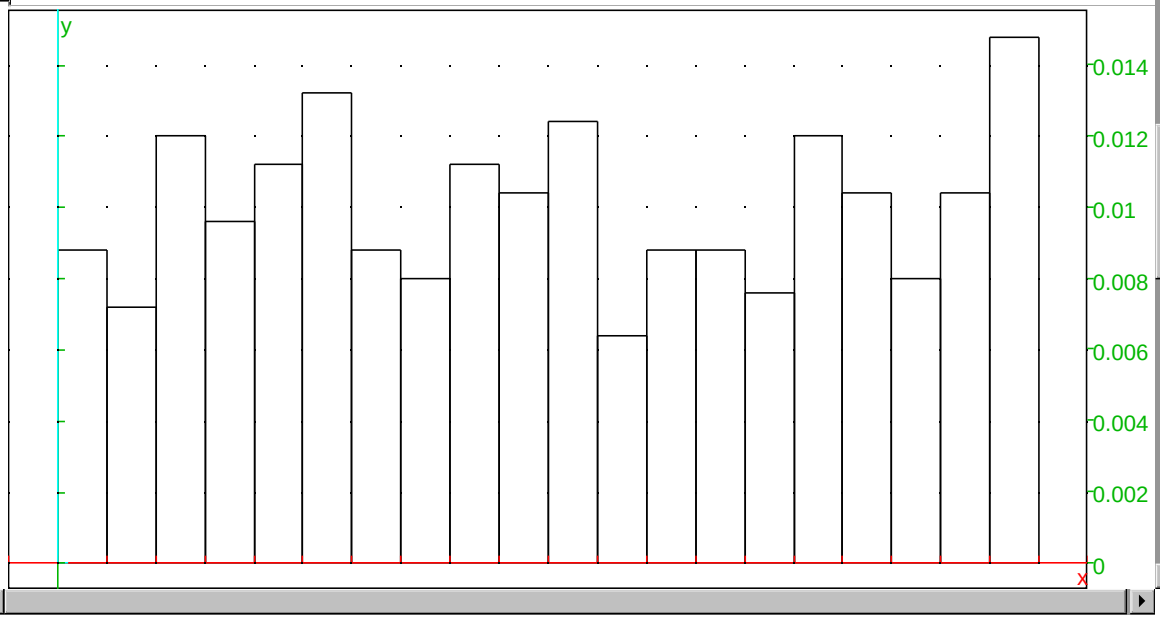
Menu

```
69 l2:=[seq(trunc(i/10^2),i=l)];
```

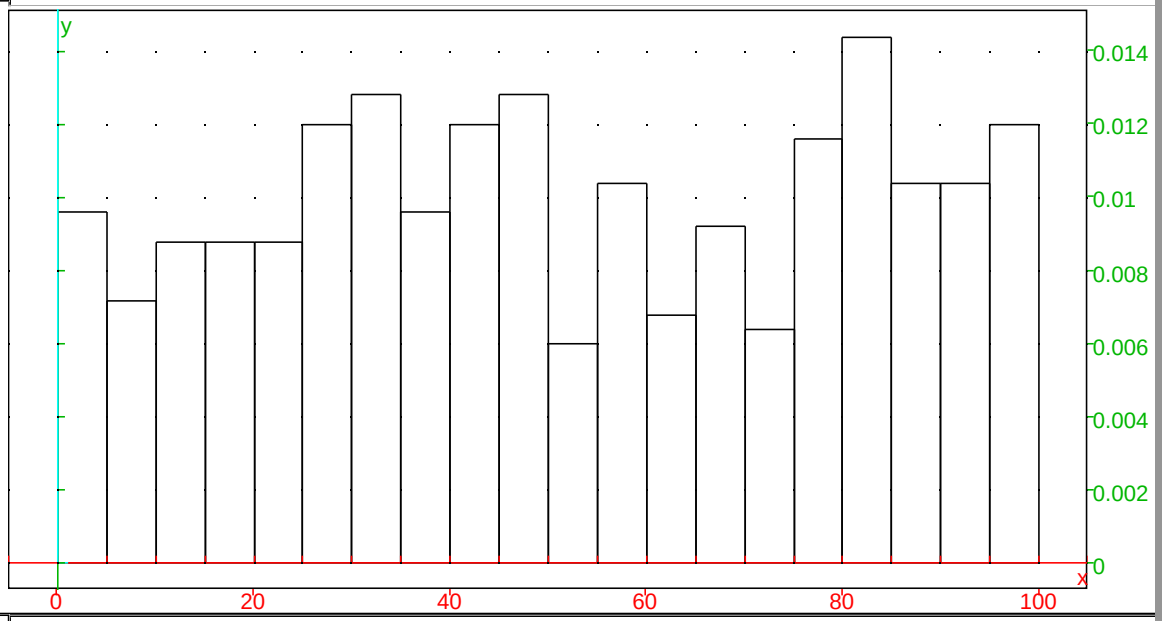
[46 96 81 97 21 82 95 67 99 40 52 39 95 22 53 1 93 25 96 86 24 83 42 41

Menu

```
70 histogram(classes(l1,0,5));
```



```
71 histogram(classes(l2,0,5));
```

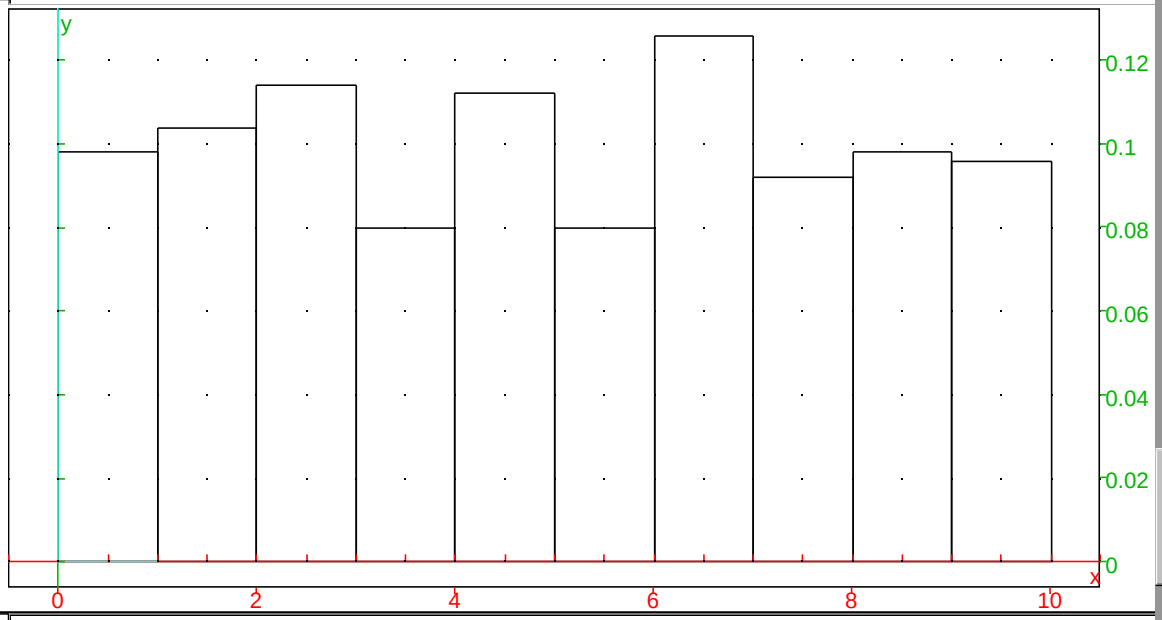



```
72 l1:=[seq(i mod 10,i=l)];
```

[3 4 4 5 2 7 0 8 9 1 8 7 3 6 6 7 6 9 6 8 7 1 9 6 8 8 4 8 2 5 6 5 5

Menu

```
73 histogram(classes(l1,0,1));
```




```
74 x:=1; a:=237;c:=54321;m:=10; l:=[];
```

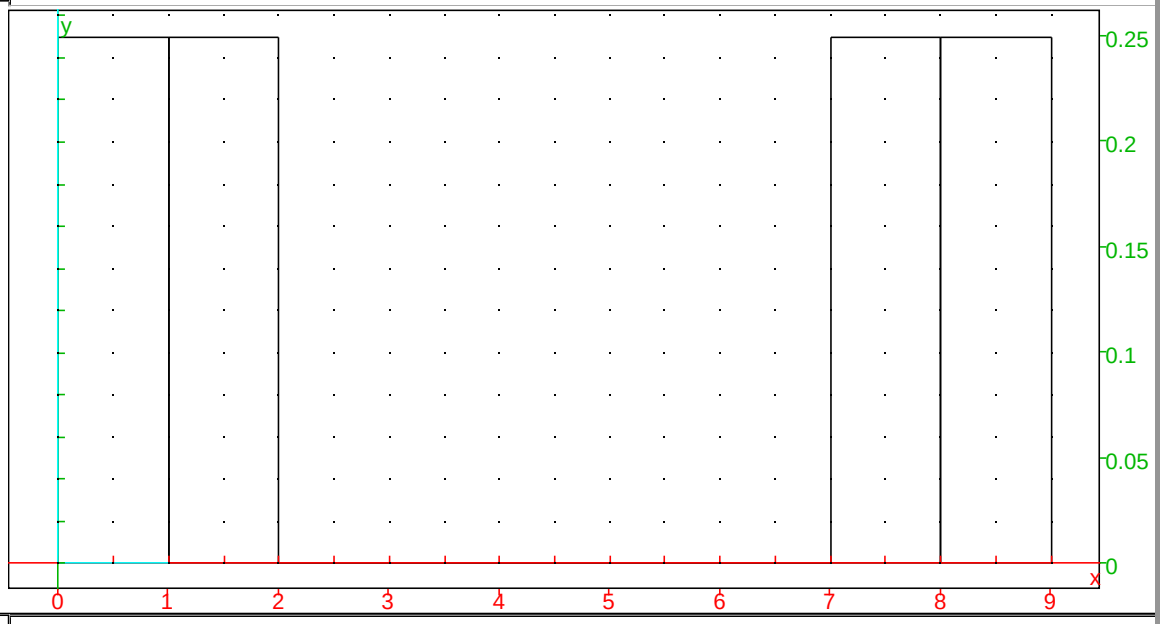
(1, 237, 54321, 10, [])

Menu

```
75 for i from 1 to 100  do x:=(a*x+c) mod m; l:=[op(l),x] ; od:
```

Done

76 histogram(classes(l,0,1));#certaines valeurs ne sont pas atteintes.



77

78 Prog Edit Add | nxt | OK | Save

```
periode:= proc(a)
x:=1;p:=1;
if (a mod 5 <> 0)  then while x>0 do p:=p+1;x:=(a*x+1) mod 125;   od; fi;
p; end  proc;
```

```
proc(a)
  x:=1;
  p:=1;
  if (irem(a,5))<>0 then
    while x>0 do
      p:=p+1;
      x:=irem(a*x+1,125);
    od;
  fi ;
  p;
```

Menu

79 periode(1),periode(5); #verification

(125, 1)

Menu

80 l:=seq(periode(i),i=2..124);max(l);

(100, 100, 50, 1, 125, 20, 100, 50, 1, 125, 100, 100, 50, 1, 125, 100, 20, 50, 1, 125, 100, 100

Menu

81 On a: x_n=(a^n-1)/(a-1) mod m, donc la periode est l'ordre de a.