

■ Generació de nombres primers

Generació de primers de longitud l amb paràmetre de seguretat k:

1. Generar un enter senar aleatori N de l dígit
2. Dividir N per primers menors que una certa fita D (taula)
3. Passar el test de Miller-Rabin amb paràmetre de seguretat k

```
lListatPrimers[D_] := Module[{l, i},
  l = Table[1, {i, 1, D}];
  l[[1]] = 0;
  For[i = 2, i <= Sqrt[D], i++, If[l[[i]] == 1, For[j = 2
  Flatten@Position[l, 1]]
```

```
lListatPrimers[500]
```

```
{2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73,
79, 83, 89, 97, 101, 103, 107, 109, 113, 127, 131, 137, 139, 149, 151,
157, 163, 167, 173, 179, 181, 191, 193, 197, 199, 211, 223, 227, 229, 233,
239, 241, 251, 257, 263, 269, 271, 277, 281, 283, 293, 307, 311, 313, 317,
331, 337, 347, 349, 353, 359, 367, 373, 379, 383, 389, 397, 401, 409, 419,
421, 431, 433, 439, 443, 449, 457, 461, 463, 467, 479, 487, 491, 499}
```

```
testBaseMillerRabin[p_, b_] := Module[{pBits, t, p0, i, x, k},
  pBits := IntegerDigits[p - 1, 2];
  i = Length[pBits]; t = 0;
  While[(i > 0) && (pBits[[i]] == 0), i--; t++];
  Print["t= ", t];
  pBits = Drop[pBits, -t];
  p0 = If[pBits == {}, 1, FromDigits[pBits, 2]];
  x = PowerMod[b, p0, p]; Print["x0= ", x];
  If [x == 1, True,
    k = 0;
    While[(x != p - 1) && (k < t - 1),
      x = Mod[x^2, p]; Print["x", k + 1, "= ", x]; k++];
    (x == p - 1)]
  ]
```

```
testBaseMillerRabin[274823765873, 2]
```

```
t= 4  
x0= 168244256840  
x1= 146362013976  
x2= 274823765872  
True
```

```
PrimeQ[274823765873]
```

```
True
```

```
testBaseMillerRabin[27482397, 87987]
```

```
t= 2  
x0= 7962540  
x1= 15960615  
False
```

```
FactorInteger[27482397]
```

```
{{3, 1}, {661, 1}, {13859, 1}}
```

```
testBaseMillerRabin[8729834573137, 7]
```

```
t= 4  
x0= 479523455020  
x1= 3598592796357  
x2= 5613933774951  
x3= 7221998329489  
False
```

```
FactorInteger[8729834573137]
```

```
{{23, 1}, {79, 1}, {4804531961, 1}}
```

```
testBaseMillerRabin[25, 2]
```

```
t= 3  
x0= 8  
x1= 14  
x2= 21  
False
```

```
testBaseMillerRabin[65, 8]
```

```
t= 6  
x0= 8  
x1= 64  
True
```

65 passa el test de Miller-Rabin en base 8, però no és primer!

```

generaPrimer[l_, k_, D_] :=
Module[{i, primers, esPrim, nPrim, base},
primers = llistatPrimers[D];
esPrim = False;
nPrim = Random[Integer, {2^(l-1), 2^l-1}];
nPrim = nPrim + 1 - Mod[nPrim, 2];
Print[nPrim];
While[! esPrim, Print["****NOU INTENT*****"];
esPrim = ! MemberQ[Table[Mod[nPrim, primers[[i]]],
{i, 1, Length[primers]}], 0];
Print["No divisible per primers petits: ",
esPrim];
i = 0;
While[(esPrim) && (i < k),
base = Random[Integer, {2, nPrim-1}];
Print["base ", base];
esPrim =
GCD[base, nPrim] == 1;
esPrim = esPrim && testBaseMillerRabin[
nPrim, base];
Print["Passa el test per
a aquesta base: ", esPrim];
i++];
nPrim = If[! esPrim, nPrim + 2, nPrim];
nPrim ];

```

Primer de 32 bits. Test de Miller-Rabin amb 4 bases. Divisibilitat per primers menors que 500

```
p = generaPrimer[32, 4, 500]
```

2171350817

****NOU INTENT*****

No divisible per primers petits: True

base 69494156


```
****NOU INTENT*****
No divisible per primers petits: True
base 1939943713
t= 6
x0= 834331602
x1= 695097994
x2= 1608032607
x3= 1262875632
x4= 1853763797
x5= 58205858
Passa el test per a aquesta base: False
****NOU INTENT*****
No divisible per primers petits: False
****NOU INTENT*****
No divisible per primers petits: False
****NOU INTENT*****
No divisible per primers petits: False
****NOU INTENT*****
No divisible per primers petits: False
****NOU INTENT*****
No divisible per primers petits: True
base 1980749158
t= 1
x0= 1
Passa el test per a aquesta base: True
base 293189215
t= 1
x0= 1
Passa el test per a aquesta base: True
base 1432816276
t= 1
x0= 2171350858
Passa el test per a aquesta base: True
base 701265756
t= 1
x0= 1
Passa el test per a aquesta base: True
```

2171350859

PrimeQ[p]

True

```
(*Començarem sempre per la base 2*)

generaPrimer[l_, k_, D_] :=
Module[{i, primers, esPrim, nPrim, base},
primers = llistatPrimers[D];
esPrim = False;
nPrim = Random[Integer, {2^(l - 1), 2^l - 1}];
nPrim = nPrim + 1 - Mod[nPrim, 2];
  Print[nPrim];
While[! esPrim, Print["****NOU INTENT*****"];
esPrim = ! MemberQ[Table[Mod[nPrim, primers[[i]]],
  {i, 1, Length[primers]}], 0];
Print["No divisible per primers petits: ",
  esPrim];
  i = 0; base = 2;
  While[(esPrim) && (i < k),
Print["base ", base];
  esPrim =
  GCD[base, nPrim] == 1;
esPrim = esPrim && testBaseMillerRabin[
  nPrim, base];
  Print["Passa el test per
  a aquesta base: ", esPrim];
  base = Random[Integer, {2, nPrim - 1}];
  i++];
nPrim = If[! esPrim, nPrim + 2, nPrim];
nPrim];
```

```
p = generaPrimer[32, 4, 500]
```

```
3425957895
****NOU INTENT*****
No divisible per primers petits: False
****NOU INTENT*****
No divisible per primers petits: False
****NOU INTENT*****
No divisible per primers petits: False
****NOU INTENT*****
No divisible per primers petits: False
****NOU INTENT*****
No divisible per primers petits: True
base 2
Passa el test per a aquesta base: False
****NOU INTENT*****
No divisible per primers petits: False
****NOU INTENT*****
No divisible per primers petits: False
****NOU INTENT*****
No divisible per primers petits: False
****NOU INTENT*****
No divisible per primers petits: True
base 2
Passa el test per a aquesta base: False
****NOU INTENT*****
No divisible per primers petits: False
****NOU INTENT*****
No divisible per primers petits: False
****NOU INTENT*****
No divisible per primers petits: True
base 2
Passa el test per a aquesta base: True
base 2362219085
Passa el test per a aquesta base: True
base 1812562390
Passa el test per a aquesta base: True
base 802311088
Passa el test per a aquesta base: True
```

3425957917

PrimeQ[p]

True

```
(*Treiem els Print *)

testBaseMillerRabin[p_, b_] := Module[{pBits, t, p0, i, x, k},
  pBits := IntegerDigits[p - 1, 2];
  i = Length[pBits]; t = 0;
  While[(i > 0) && (pBits[[i]] == 0), i--; t++];
  pBits = Drop[pBits, -t];
  p0 = If[pBits == {}, 1, FromDigits[pBits, 2]];
  x = PowerMod[b, p0, p];
  If[x == 1, True,
    k = 0;
    While[(x != p - 1) && (k < t - 1),
      x = Mod[x^2, p]; k++];
    (x == p - 1)]
]
```

■ Generació de primers

No generem la taula de primers petits cada vegada, la posem com a paràmetre de la funció.

```

genPrimer[l_, k_, taula_] :=
Module[{i, esPrim, nPrim, base},
esPrim = False;
nPrim = Random[Integer, {2^(l - 1), 2^l - 1}];
nPrim = nPrim + 1 - Mod[nPrim, 2];
While[! esPrim,
esPrim =
! MemberQ[Table[Mod[nPrim, taula[[i]]],
{i, 1, Length[taula]}], 0];
i = 0; base = 2;
While[(esPrim) && (i < k),
esPrim =
GCD[base, nPrim] == 1;
esPrim = esPrim && testBaseMillerRabin[
nPrim, base];
base = Random[Integer, {2, nPrim - 1}];
i++];
nPrim = If[! esPrim, nPrim + 2, nPrim];
nPrim ];

```

```

ta = llistatPrimers[10000];

```

```

Length[ta]

```

```

1229

```

```

genPrimer[512, 5, ta] // Timing

```

```

{0.375 Second,
12277227751202256452516297185854029281620800643076360616184929168448353\
7653425608089534875557234699383999830380808615436699397334624877985565\
15982332155089}

```

```
genPrimer[512, 10, ta] // Timing
```

```
{1.406 Second,  
74501838977809868167058262796940718527011114655097194069479211225347055\  
3484603330702326075833723324235516065860742160396023898159594931894278\  
5807471013857}
```

```
genPrimer[1024, 5, ta] // Timing
```

```
{4.797 Second,  
13204089315712415175628453958374511196810384309066749133436825458578908\  
8824301688859104497394825504707821052776307605206819763729924175544949\  
7841019093080014217235745627772078989907655434358257145268910122614329\  
3722091304500785545408810692383856727628183034506665946657736013654227\  
8512894469899797480527973991}
```

```
genPrimer[1024, 10, ta] // Timing
```

```
{4.359 Second,  
17290864336021827990342194841399252303057964701485640595827134773038820\  
0313464496109628849941860210001604017364882567320628374693125620314073\  
1997013218301541412384609013702830801860667537222475372816279143924183\  
7087653483226845727342803133258602704672319901806894179829621396765875\  
0743461230864158312149329509}
```

```
p = genPrimer[1024, 5, ta]
```

```
130420696721734481340099382598026148257665420136690233171345417742591665\  
90490394479568639524073963574021293558085185321893770512959693878339051\  
87162389677372196118546696982114025615549423212908919512014982972979803\  
61826093688836254954408642896755671313750040309131741674695220426490777\  
146597161637254115127123
```

```
q = genPrimer[1024, 5, ta]
```

```
159741687885305829605797400625779582241887010994788506325366694286003930\  
42668406816541856705542432625979550676284837502441250681842689775359077\  
08788035650405448898932578394417464480191612765997486159438428623975213\  
93323911824287217560921692505643258967076320798240810630584769462022315\  
153629576269758442095147
```

$n = p * q$

```
208336222295074387243387425063787360425288661104899018401043620936621989:  
54245032563647011692416344760995186925985516661207311919178121170376001:  
10398933283558941012785076699016245720092169897473175898370914991942880:  
05359077323959524935352380477341184018490636352874515440570340954333342:  
71586009087912506773740414121558631038485047490348774652867122199759254:  
59445230688501584925713827064973312665341008228227538392521050538923358:  
33781117345432537426494671705144786362156955901627410924064344886161500:  
07000256439799292031585883738693378453126039407732814657594009288324908:  
036600764597386277159035211982929149714366372081
```

Length@IntegerDigits[n, 2]

2048

FactorInteger[n]

\$Aborted