...A Cifra RSA...


```
> restart:
> with(numtheory):
>
```

"Pequeno" Teorema de Fermat...

```
> p:=341:isprime(p);
  2&^(p-1) mod p;
>
```

18769...341...

```
>
```

Velocidades...

Fazer n=101, 137...

```
> n:=2^101-1;
  floor(evalf(log(n)/log(10)))+1;
> st:=time():
  isprime(n);
  time()-st;
> st:=time():
  ifactor(n);
  time()-st;
> st:=time():
  2 &^ 5 mod 7;
  time()-st;
>
```

Exponenciacao modular...

```
> expmod:=proc(a,e,n)
    local x,y,s:
    x:= a: y:=1: s:=e:
    while s<>0 do
      if s mod 2=1 then y:=y*x mod n: fi:
      x:=x*x mod n:
      s:=(s-(s mod 2))/2:
    od:
  return y:
  end proc:
> st:=time():
  expmod
```

```
   (232198691876176891982987628972687268727689716987198719879187981
   92879861,
   187976983798739892869287917099387938798392928792879287298729
   170190179387872681257146536564249874397398793873987392792879287
   928792894646484876238994879387938793879287928729879279827928794
   879392892762187681765187767 61,109982687268726897628726872 3);
   time()-st;
>
```

Gerar primos aleatoriamente...

```
> # Um exemplo do uso de procedimento "rand" para gerar um dado...
  dado:=rand(1..6):
  dado();
> # como gerar um primo aleatorio com b bits...
  b:=1024:
  rd:=rand(2^(b-1)..2^b-1):
  n:=rd():
  p:=nextprime(n):
  printf("O número:\n %A\n\né um primo com %A bits.\n",p,floor
  (evalf(log(p)/log(2)))+1):
> # numero de algarismos de p...
  floor(evalf(log(p)/log(10)))+1;
>
>
```

Construcao de uma cifra RSA...

```
> # dois primos aleatorios com b bits...
  b:=1024:
  rd:=rand(2^(b-1)..2^b-1):
  n:=rd(): p:=nextprime(n):
  n:=rd(): q:=nextprime(n):
  printf("p=%A\nq=%A\n\n",p,q):
> # numero de algarismos de p e q, respectivamente...
  floor(evalf(log(p)/log(10)))+1,floor(evalf(log(q)/log(10)))+1;
> n:=p*q: printf("n=%A",n):
> # gera c tal que (c,phi(n))=1, aleatoriamente...
  g:=0:
  while g<>1 do
  c:=rd();g:=igcd(c,(p-1)*(q-1)):od:
  printf("c=%A",c);
> # calculo do inverso de c mod phi(n)...
  igcdex(c,(p-1)*(q-1),'d','z'):
  printf("d=%A\n",d):
> # uma cifra RSA...
  printf("n=%A\n c=%A\n d:=%A",n,c,d):
```

```
>
>
```

Codificacao...

```
> with(StringTools):
> Mg:="A CRIPTOGRAFIA E UM ASSUNTO VERDADEIRAMENTE APAIXONANTE":
  #Mg:="A TEORIA DOS NUMEROS E UMA DAS AREAS MAIS BELAS DE TODO O
  CONHECIMENTO HUMANO":
  Mg:=SubstituteAll(Mg," ",""):
  Mg:=convert(Mg,decimal,36);
>
> C:=Mg &^c mod n;
>
>
```

Descodificacao...

```
> Md:=C &^d mod n:
  Md:=convert(Md,base,36): m:=nops(Md):
  Morig:="":
  for i from 0 to m-1 do
   Morig:=cat(Morig,convert([Md[m-i]+55],bytes)):
  od:
  Morig;
>
>
```

Os números RSAs...

```
> RSA129:=
  114381625757888867669235779976146612010218296721242362562561842935706935245733897830597123563958705058989075147599290026879543541:
  ceil(evalf(ln(RSA129)/ln(2),200));
  p1:=
  3490529510847650949147849619903898133417764638493387843990820577:
  p2:=
  3276913299326670954996198819083446141317764296799294253979828853 3:
> # $10,000...
  RSA576:=
  1881988129206079638386972394616504398071635633794173827007633564229888597152346654853190606065047430453173880113033967161996923212057340318795506569962213051687593076502570 59:
  ceil(evalf(ln(RSA576)/ln(10),200));
> # $20,000...
```

```
RSA640:=
3107418240490043721350750035888567930037346022842727545720161948
8232064405180815045563468296717232867824379162728380334154710731
0850191954852900733772482278352574238645401469173660247765234660
9:
ceil(evalf(ln(RSA640)/ln(10),200));
> # $30,000...
RSA704:=
7403756347956171282804679609742957314259318888923128908493623263
8972765034028266276891996419625117843995894330502127585370118968
0982867331732731089309005525051168770632990723963807867100860969
62537934650563796359:
ceil(evalf(ln(RSA704)/ln(10),200));
> # $50,000...
RSA768:=
1230186684530117755130494958384962720772853569595334792197322452
1517264005072636575187452021997864693899564749427740638459251925
5732630345373154826850791702612214291346167042921431160222124047
9274737794080665351419597459856902143413:
ceil(evalf(ln(RSA768)/ln(2),200));
> # $75,000...
RSA896:=
4120234369866595438555313653325759481798116998443279828454556264
3387644556524842619809887042316184187926142024718886949256093177
6375033421130982397485150944909106910269861031862704114880866970
5649029036536588674337317208131041051908642547932826013912576240
33946373269391:
ceil(evalf(ln(RSA896)/ln(2),200));
> # $100,000...
RSA1024:=
1350664108659952233496032162788059699388814756056670275244851438
5152651060485953383394028715057190944179820728216447155137368041
9703964191743046496589274256239341020864383202110372958725762358
5096431105640735015081875106765946292055636855294752135008528794
1637732853390610975054433499981115005697723689092756 3:
ceil(evalf(ln(RSA1024)/ln(2),200));
> # $150,000...
RSA1536:=
1847699703211741474306835620200164403018549338663410171471785774
9106516967111612498593376843054357445856160615445717940522297177
3252466096064694607124962372044202226975675668737842756238950876
4678440933285157496578843415088475528298186726451339863364931908
0846719904318743812833635027954702826532978029349161558118810498
4490831954500984839377522725705257859194499387007369575568843693
3812779613089230392569695253261620823676490316036551371447913932
347169566988069:
ceil(evalf(ln(RSA1536)/ln(2),200));
```

```
> # $200,000...
  RSA2048:=
  25195908475657893494027183240048398571429282126204032027777137836043662020707595556264018525880784406918290641249515082189298559149176184502808489120072844992687392807287776735971418347270261896375014971824691165077613379859095700097330459748808428401797429100642458691817195118746121515172654632282216869987549182422433637259085141865462043576798423387184774447920739934236584823824281198163815010674810451660377306056201619676256133844143603833904414952634432190114657544454178424020924616515723350778707749817125772467962926386356373289912154831438167899885040445364023527381951378636564391212010397122822120720357:
  ceil(evalf(ln(RSA2048)/ln(10),200));
> st:=time():isprime(RSA2048);time()-st;
>
>
```

Novamente o "pequeno" Teorema de Fermat...

```
> n:=RSA2048:
  2&^(n-1) mod n;
>
>
```

Raizes Primitivas...

```
> # calcula a menor raiz primitiva mod p...
  p:=41:
  primroot(p);
> # calcula a menor raiz primitiva mod p maior que a...
  p:=41:
  a:=6:
  primroot(a,p);
> # calculo de todas as phi(p-1) raizes primitivas mod p...
  p:=41:
  g:=0:
  ct:=0:
  while primroot(g,41)<>FAIL do:
    g:=primroot(g,41): ct:=ct+1:
    printf("(%A) %A\n",ct,g):
  od:
  printf("\nHá %A raízes primitivas mod %A",phi(p-1),p);
>
>
```

Protocolo ElGamal...

```
> # um primo aleatorio com b bits...
```

```
   b:=128:
   rd:=rand(2^(b-1)..2^b-1):
   n:=rd():
   p:=nextprime(n):
   printf("O número: %A é um primo com %A bits.\n",p,floor(evalf
   (log(p)/log(2)))+1):
> # Uma raiz primitiva mod p...
   alpha:=primroot(p);
> # Um número aleatorio do conjunto {1,2,...,p-2}
   rd:=rand(1..p-2):
   a:=rd();
> b:=alpha &^ a mod p;
> # a chave pública:
   printf("p=%A\nalpha=%A\nb=%A",p,alpha,b);
> #
   # número de apostas no euromilhões
   # (se as escolhas forem 2 de 9 estrelas e 5 de 50 números...)
   #
   binomial(9,2)*binomial(50,5);
> # a mensagem secreta, como elemento do conjunto {0,1,2,...,p-1},
   e um número aleatorio do conjunto {1,2,...,p-2}...
   m:=76275360;
   k:=rd();
> # As duas partes do criptograma
   gamma1:=alpha &^ k mod p;
   delta:= m*b &^ k mod p;
> # decifração...
   gamma1 &^ (-a) * delta mod p;
>
>
```