In [12]:
```
p=next_prime(2345820)
```

In [13]:
```
p
```

Out[13]:
```
2345831
```

In [21]:
```
g=13
for i in range(1,p):
   if(power_mod(g,i,p) == 1):
      print(i); break;
```

```
2345830
```

In [22]:
```
#como el orden de 13 es p-1, podemos usarlo como generador de (Z_p)^
*
```

In [23]:
```
#Alice y Bob se ponen de acuerdo en usar p y en g
```

In [24]:
```
a=1700432 #Alice escoge su número secreto
```

In [25]:
```
b=123456 #Bob escoge su número secreto
```

In [26]:
```
A=g^a % p # es esto pero mejor hacerloo así
```

In [28]:
```
A=power_mod(g,a,p); A #Alice envia A a Bob
```

Out[28]:
```
1151440
```

In [29]:
```
B=power_mod(g,b,p); B #Bob envia B a Alice
```

Out[29]:
```
1160264
```

In [31]:
```
K1=power_mod(B,a,p); K1 #Alice calcula la clave gracias a lo que ha
mandado Bob y a su número secreto a
```

Out[31]:
```
1212772
```

In [32]:
```
K2=power_mod(A,b,p); K1 #Bob calcula la clave gracias a lo que ha ma
ndado Alice y a su número secreto b
```

Out[32]:
```
1212772
```

In [33]:
```
#Y las dos clave coinciden
```

In [ ]:
```
#Intento de Eve de romper el sistema
```

In [36]:
```
for i in range(1,p):
   if(power_mod(g,i,p) == A):
      print(i); aEve=i; break;
```

```
1700432
```

In [37]:
```
for i in range(1,p):
   if(power_mod(g,i,p) == B):
      print(i); bEve=i; break;
```

```
123456
```

In [50]:
```
g^aEve %p == A % p; g^bEve %p == B % p
```

Out[50]:
```
True
```

In [43]:
```
a == aEve
```

Out[43]:
```
True
```

In [51]:
```
b == bEve
```

Out[51]:
```
True
```

In [62]:
```
K3=power_mod(g,(aEve*bEve % (p-1)),p)
```

In [63]:
```
K3
```

Out[63]:
```
1212772
```

In [64]:
```
K1==K3
```

Out[64]:
```
True
```

In [65]:
```
#Lo ha roto porque ha podido calcular el logartimo
```

In [ ]:
```

```