Exercice 1: Une variante de RSA - correction

Toutes les commandes Sagemath sont autorisées pour cet exercice.

In [8]:
```
def lettreanombre(l):
    if l==' ':
        return 0
    else:
        return ord(l)-96

def nombrealettre(n):
    if n==0:
        return ' '
    else:
        return chr(n+96)

def decode(n):
    a=n
    l="";
    while (a>0):
        l=nombrealettre(a%27)+l;
        a=a//27
    return l

def encode(l):
    i=0;
    n=0;
    while (i<len(l)):
        n=lettreanombre(l[i])+27*n
        i=i+1
    return n
```

Au laboratoire de mathématiques, tous les messages que s'écrivent les chercheurs sont codés suivant le protocole suivant:

- on a choisi 3 entiers premiers p_1, p_2 et p_3 et en les multipliant, on a créé une clé publique
$$n = 313297851918307610502911324433872045797703258998425302542562974 63$$
- les messages alphanumériques sont transformés en un grand entier M (comme dans le tp2) via la fonction `encode()` donnée ci-dessus;
- Le message est crypté sous la forme $C = M^e \bmod n$ grâce à la clé publique de cryptage
$$e = 10616692270673$$

In [2]:
```
n=313297851918307610502911324433872045797703258998425302542562974 63
e=10616692270673
```

Question a)

Transmettre de façon cryptée le message suivant: "dans ce cours on apprend des choses utiles"

In [3]:
```
message="dans ce cours on apprend des choses utiles"
```

In [29]:
```
M=encode(message);print 'M=',M
print decode(M)
C=Mod(M,n)^e;print 'C=',C
```

```
M= 207864186116434408103579562282567250912237323718764267074756402 71
utrfqwbeutbiqjvdouvwhgebj lxuzdllvdxwyqaeechs
C= 129782391931533930118897587876845762913632563735855960826274933 84
```

In [16]:
```
print 'message crypté alphabétique :',decode(129782391931533930118897587876845762913632563735855960826274933 84)
```

```
message crypté alphabétique : moxxfkwkkcfvwcgaqeayqksfrpiwnhmiygmgwaacspfpc
```

Question b)

Ce protocole n'est pas sûr. Vous avez intercepté un message. Décodez le, (en laissant apparents vos calculs intermédiaires)!

In [3]:
```
message="utrfqwbeutbiqjvdouvwhgebj lxuzdllvdxwyqaeechs"
```

In [4]:
```
phi=euler_phi(n);print phi
```

```
313297851918307610502520404469140332640143721639961143906736680000
```

In [6]:
```
print xgcd(e,phi)
f=xgcd(e,phi)[1]  #f est l'inverse de e modulo phi
print 'vérification : mod(e,phi)*f =',mod(e,phi)*f
```

```
(1, 450759625701593021752748328520732858812045473795675654545755073 7, -1527484534227)
vérification : mod(e,phi)*f = 1
```

In [9]:
```
C=encode(message);print 'C=',C
N=mod(C,n)^f;print 'N=',N
print 'vérification N^e = C mod n ?',N^e==mod(C,n)
```

```
C= 207864186116434408103579562282567250912237323718764267074756402 71
N= 944105305276458624267599313753120739638119592112162612871922 74
vérification N^e = C mod n ? True
```

In [25]:
```
print decode(944105305276458624267599313753120739638119592112162612871922 74)
```

```
bravo vous pouvez passer a la suite du sujet
```