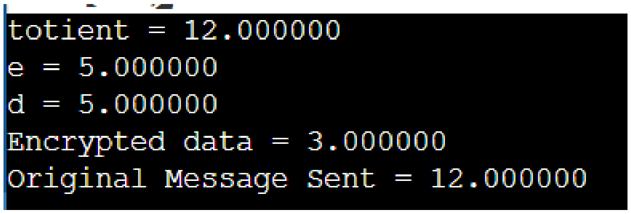
# Implementation of RSA

```
#include<stdio.h>
#include<math.h>
//to find gcd
int gcd(int a, int h)
{
    int temp;
    while(1)
    {
        temp = a%h;
        if(temp==0)
        return h;
        a = h;
        h = temp;
    }
}
int main()
{
    //2 random prime numbers
    double p = 3;
    double q = 7;
    double n=p*q;
    double count;
    double totient = (p-1)*(q-1);

    //public key
    //e stands for encrypt
    double e=2;
    //for checking co-prime which satisfies e>1
    while(e<totient){
    count = gcd(e,totient);
    if(count==1)
        break;
    else
        e++;
    }
    //private key
    //d stands for decrypt
    double d;
    //k can be any arbitrary value
    double k = 2;
    //choosing d such that it satisfies d*e = 1 + k * totient
    d = (1 + (k*totient))/e;
    double msg = 12;
    double c = pow(msg,e);
    double m = pow(c,d);
    c=fmod(c,n);
    m=fmod(m,n);
    printf("Message data = %lf",msg);
    printf("\np = %lf",p);
    printf("\nq = %lf",q);
    printf("\nn = pq = %lf",n);
    printf("\ntotient = %lf",totient);
    printf("\ne = %lf",e);
    printf("\nd = %lf",d);
    printf("\nEncrypted data = %lf",c);
    printf("\nOriginal Message Sent = %lf",m);
    return 0;
}
```

Output



```
totient = 12.000000
e = 5.000000
d = 5.000000
Encrypted data = 3.000000
Original Message Sent = 12.000000
```