EXPERIMENT-06

Aim:Write a Program to Implement and analyze RSA cryptosystem.

Roll No.	70
Name	MAYURI SHRIDATTA YERANDE
Class	D15-B
Subject	Internet Security Lab
LO Mapped	LO1: To apply the knowledge of symmetric cryptography to implement classical ciphers.

AIM: Write a Program to Implement and analyze RSA cryptosystem.

THEORY:

RSA algorithm is an asymmetric cryptography algorithm. Asymmetric actually means that it works on two different keys i.e. Public Key and Private Key. As the name describes, the Public Key is given to everyone and the Private key is kept private.
The idea! The idea of RSA is based on the fact that it is difficult to factorize a large integer. The public key consists of two numbers where one number is multiplication of two large prime numbers. And private keys are also derived from the same two prime numbers. So if somebody can factorize the large number, the private key is compromised. Therefore encryption strength totally lies on the key size and if we double or triple the key size, the strength of encryption increases exponentially. RSA keys can be typically 1024 or 2048 bits long, but experts believe that 1024 bit keys could be broken in the near future. But till now it seems to be an infeasible task.
>> Generating Public Key :Select two prime no's. Suppose P = 53 and Q = 59.

- Now First part of the Public key : n = P*Q = 3127.
- We also need a small exponent say e :
- But e Must be:- An integer., Not be a factor of n.
- 1 < e < Φ(n) [Φ(n) is discussed below],
- Our Public Key is made of n and e

>> Generating Private Key :We need to calculate Φ(n) :
Such that Φ(n) = (P-1)(Q-1)

- so, Φ(n) = 3016
- Now calculate Private Key, d : d = (k*Φ(n) + 1) / e for some integer k
- For k = 2, value of d is 2011.
- Now we are ready with our – Public Key (n = 3127 and e = 3) and Private Key(d = 2011)

  Now we will encrypt “HI” :Convert letters to numbers : H = 8 and I = 9
- Thus Encrypted Data c = 89e mod n.
- Thus our Encrypted Data comes out to be 1394

Now we will decrypt 1394 :

Decrypted Data = cd mod n.

- Thus our Encrypted Data comes out to be 89
- 8 = H and I = 9 i.e. "HI".

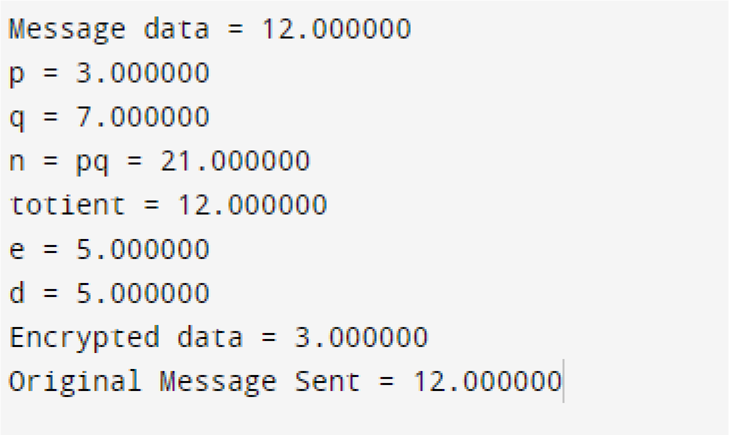
IMPLEMENTATION:

CODE:

```
#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()
{

   double p = 3;
   double q = 7;
   double n=p*q;
   double count;
   double totient = (p-1)*(q-1);

   //public key
   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:



```
Message data = 12.000000
p = 3.000000
q = 7.000000
n = pq = 21.000000
totient = 12.000000
e = 5.000000
d = 5.000000
Encrypted data = 3.000000
Original Message Sent = 12.000000
```

CONCLUSION: We have successfully implemented and analyzed the RSA cryptosystem.