LAB ASSESSMENT 4

Name: Charan Lalchand Soneji

Registration number: 17BCE2196

Faculty: Prof Marimuthu K.

Course Title: Cryptography fundamentals

Slot: L51+ L52

7. To implement the RSA Public key cryptosystem
9. To implement the Diffie-Hellman Key exchange algorithm

Code (C code)f

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#include<math.h>
#include<string.h>
long int p,q,n,t,flag,e[100],d[100],temp[100],j,m[100],en[100],i;
char msg[100];
int prime(long int);
void ce();
long int cd(long int);
void encrypt();
void decrypt();
long long int power(long long int a, long long int b, long long int
P)
{
      if (b == 1)
            return a;
      else
            return (((long long int)pow(a, b)) % P);
}


int main() {
      printf("----------------------Welcome 17BCE2196---------------
------------\n");
```

```
	int o;
	printf("\nPlease choose the encryption algorithm");
	printf("\n1. RSA Public Key Cryptosystem");
	printf("\n2. Diffie Hellman Key Exchange Algorithm");
	printf("\n3. Exit\n");
	scanf("\n%d",&o);
	switch(o){
		case 1:
			rsam();
			break;
		case 2:
			printf("The Secret keys for the common example of
Alice and Bob shall now be calculated by the method of Diffie
Hellman Algorithm\n");
			diffie();
			break;
		case 3:
			break;
}
}
void diffie(){
	long long int P, G, x, a, y, b, ka, kb;
	printf("Enter the value of P(Prime number):");
	scanf("%d",&P);
	printf("The value of P : %lld\n", P);
	printf("Enter the value of G(Prime number):");
	scanf("%d",&G);
	printf("The value of G : %lld\n", G);
	printf("Enter the private key chosen by Alice\n");
	scanf("%d",&a);
	printf("The private key a for Alice : %lld\n", a);
	x = power(G, a, P);
	printf("Enter the private key chosen by Bob\n");
	scanf("%d",&b);
	printf("The private key b for Bob : %lld\n", b);
	y = power(G, b, P);
```

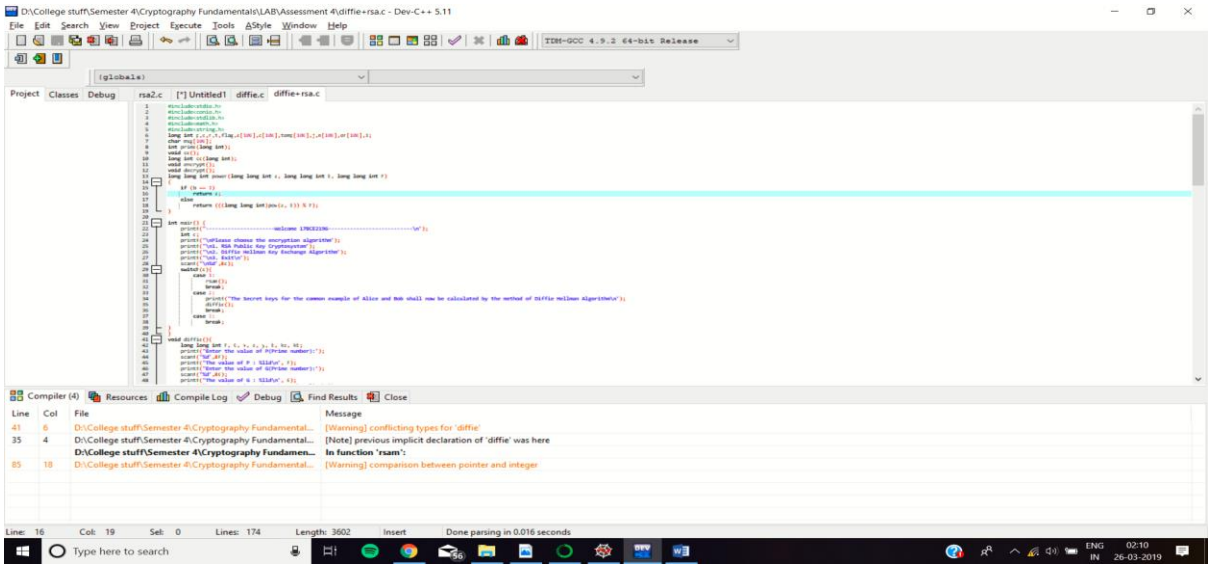
```
      ka = power(y, a, P);
      kb = power(x, b, P);
      printf("The keys generated have now been exchanged\n");
      printf("Secret key for the Alice is : %lld\n", ka);
      printf("Secret Key for the Bob is : %lld\n", kb);
}


int rsam(){
      printf("Encryption and Decryption in RSA algorithm shall now
be performed\n");
      printf("Enter the value of p(First Prime number):");
      scanf("%d",&p);
      flag=prime(p);
      if(flag==0) {
            printf("Invalid Input");
            getch();
            exit(1);
      }
      printf("Enter the value of q(Second prime number):");
      scanf("%d",&q);
      flag=prime(q);
      if(flag==0||p==q) {
            printf("Invalid Input\n");
            getch();
            exit(1);
      }
      printf("Enter Message");
      fflush(stdin);
      scanf("%s",msg);
      for (i=0;msg[i] !=NULL;i++)
            m[i]=msg[i];
      n=p*q;
      t=(p-1)*(q-1);
      ce();
      printf("Possible values of e and d are:");
```

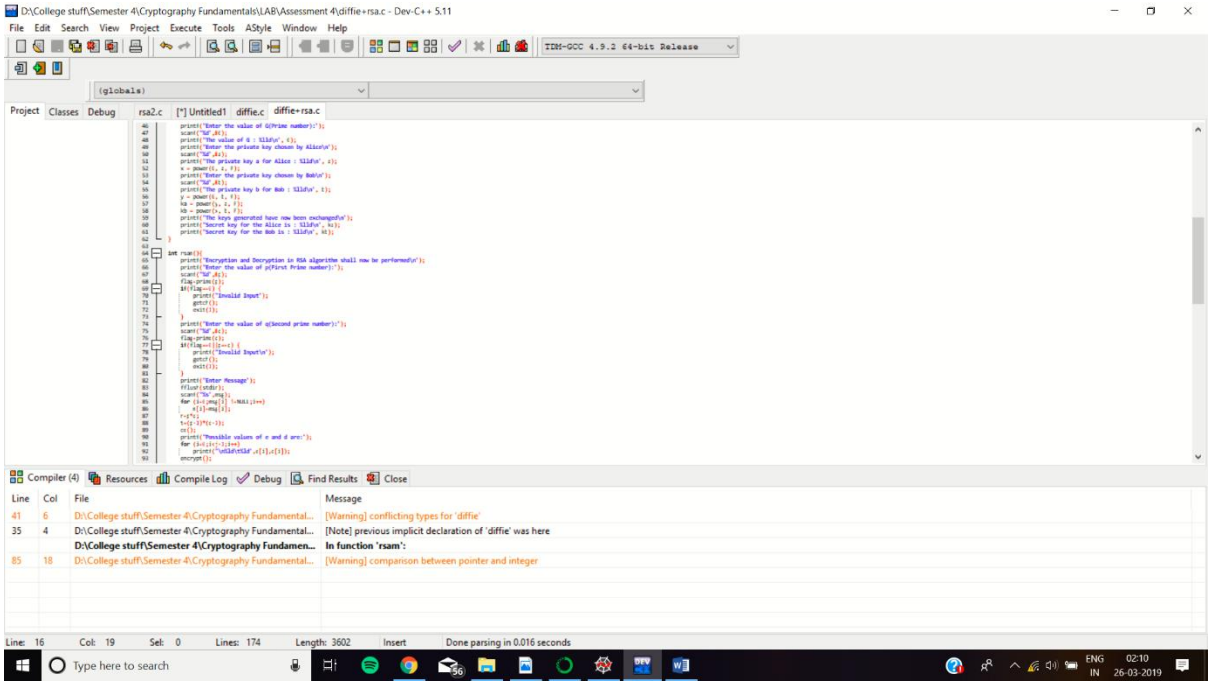
```
      for (i=0;i<j-1;i++)
            printf("\n%ld\t%ld",e[i],d[i]);
      encrypt();
      decrypt();
      getch();
}
int prime(long int pr) {
      int i;
      j=sqrt(pr);
      for (i=2;i<=j;i++) {
            if(pr%i==0)
                return 0;
      }
      return 1;
}
void ce() {
      int k;
      k=0;
      for (i=2;i<t;i++) {
            if(t%i==0)
                continue;
            flag=prime(i);
            if(flag==1&&i!=p&&i!=q) {
                 e[k]=i;
                 flag=cd(e[k]);
                 if(flag>0) {
                       d[k]=flag;
                       k++;
                 }
                 if(k==99)
                         break;
            }
      }
}
long int cd(long int x) {
```

```
	long int k=1;
	while(1) {
		k=k+t;
		if(k%x==0)
		    return(k/x);
	}
}
void encrypt() {
	long int pt,ct,key=e[0],k,len;
	i=0;
	len=strlen(msg);
	while(i!=len) {
		pt=m[i];
		pt=pt-96;
		k=1;
		for (j=0;j<key;j++) {
			k=k*pt;
			k=k%n;
		}
		temp[i]=k;
		ct=k+96;
		en[i]=ct;
		i++;
	}
	en[i]=-1;
	printf("\nThe Encrypted message is:");
	for (i=0;en[i]!=-1;i++)
	printf("%c",en[i]);
}
void decrypt() {
	long int pt,ct,key=d[0],k;
	i=0;
	while(en[i]!=-1) {
		ct=temp[i];
		k=1;
```

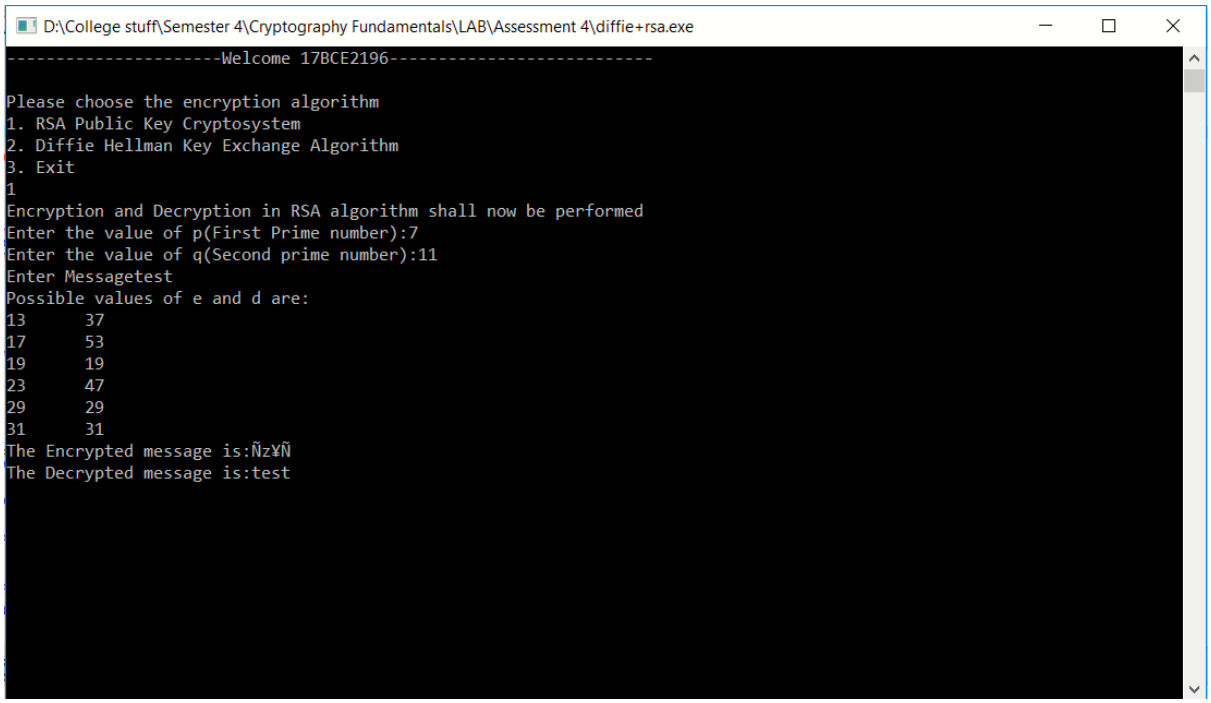
```
            for (j=0;j<key;j++) {
                  k=k*ct;
                  k=k%n;
            }
            pt=k+96;
            m[i]=pt;
            i++;
      }
      m[i]=-1;
      printf("\nThe Decrypted message is:");
      for (i=0;m[i]!=-1;i++)
            printf("%c",m[i]);
}
```

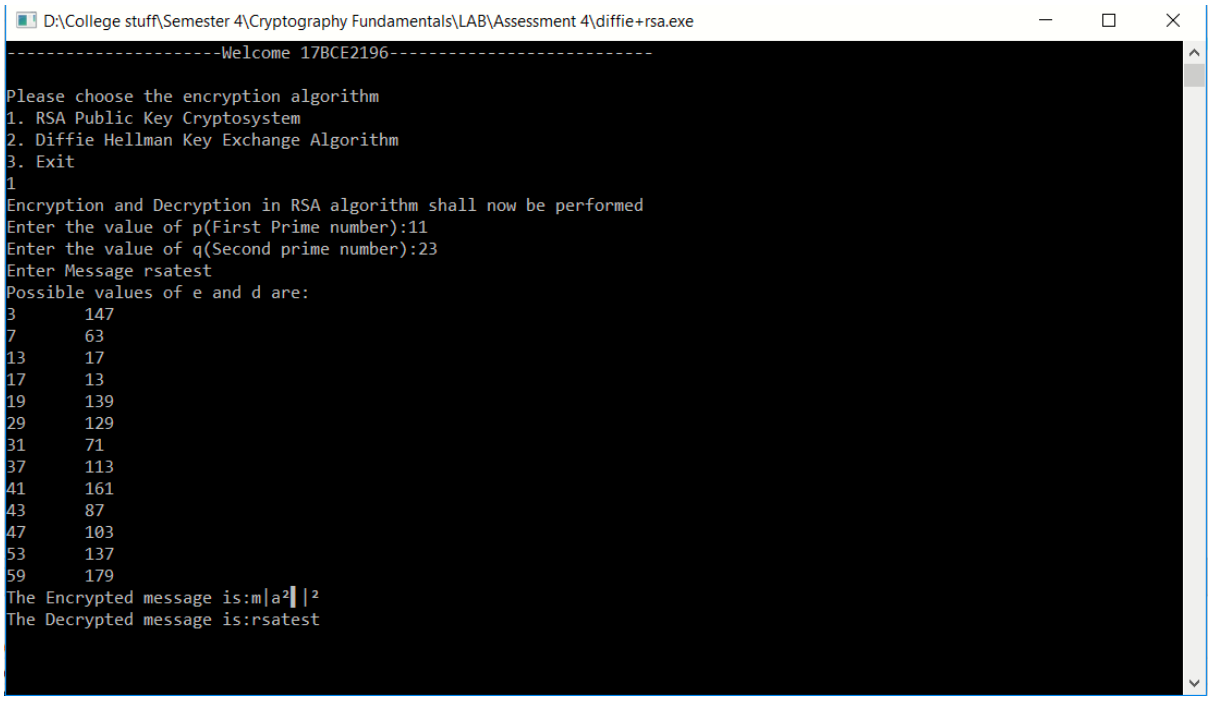
CODE SNIPPETS





OUTPUT SNIPPET





8. To implement the ElGamal Public key cryptosystem

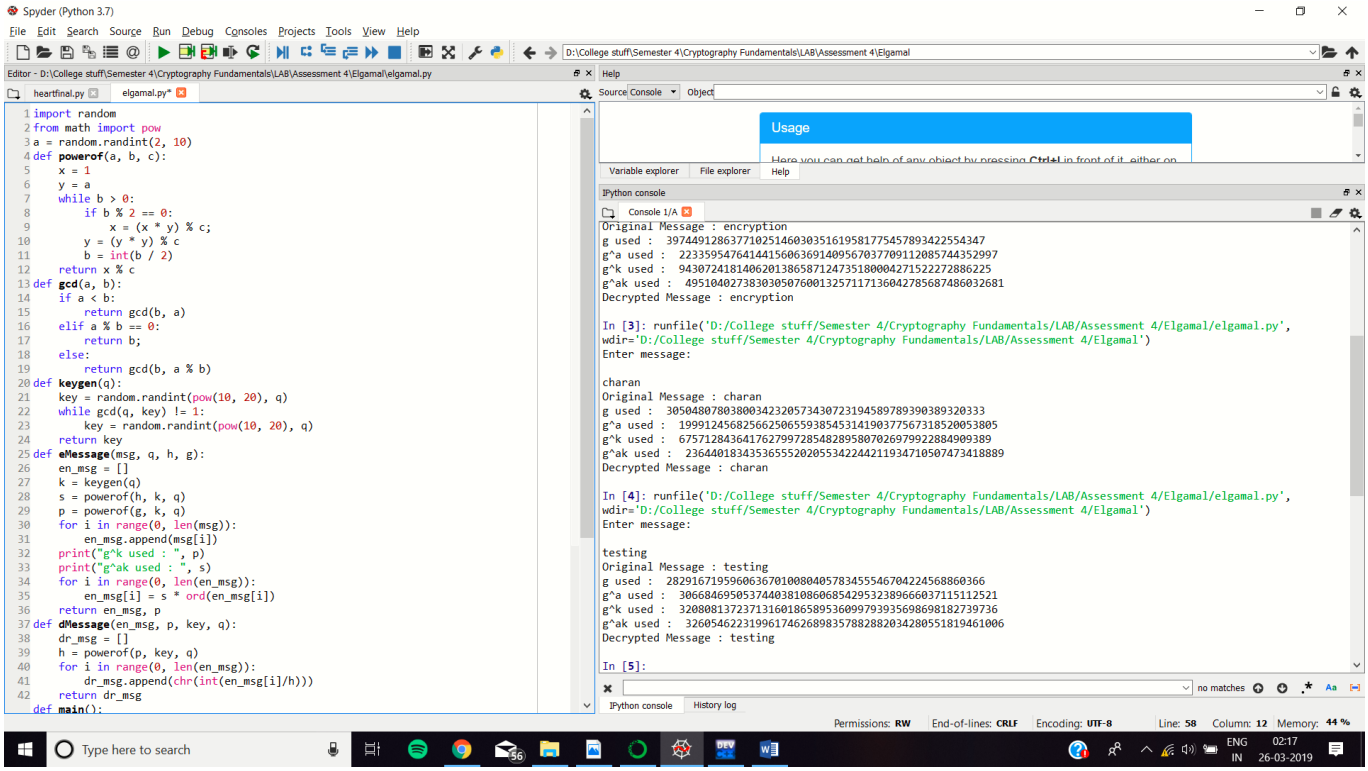
CODE (In Python script-In Spyder Enviroment)

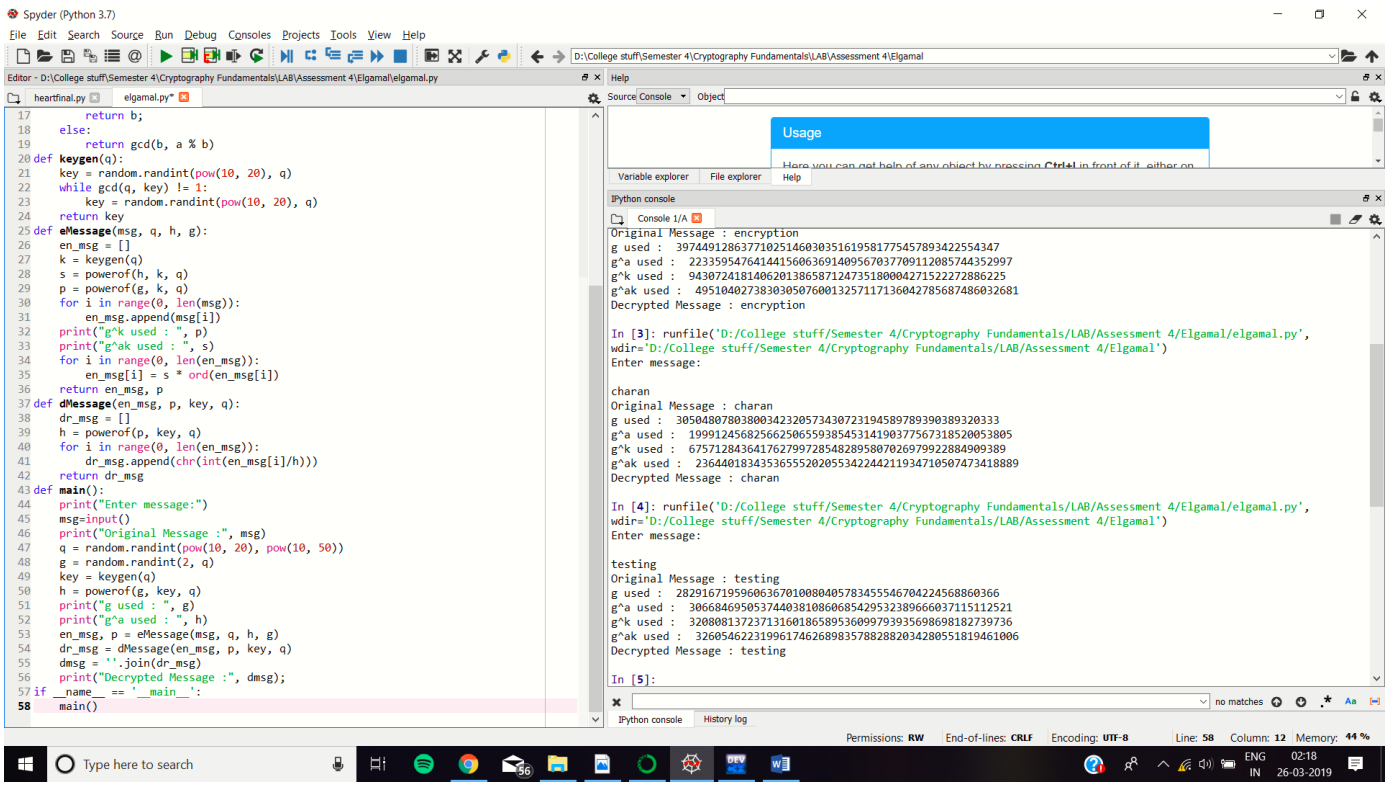
```
import random

from math import pow

a = random.randint(2, 10)

def powerof(a, b, c):

    x = 1
```

```
    y = a
    while b > 0:
        if b % 2 == 0:
            x = (x * y) % c;
        y = (y * y) % c
        b = int(b / 2)
    return x % c
def gcd(a, b):
    if a < b:
        return gcd(b, a)
    elif a % b == 0:
        return b;
    else:
        return gcd(b, a % b)
def keygen(q):
    key = random.randint(pow(10, 20), q)
    while gcd(q, key) != 1:
        key = random.randint(pow(10, 20), q)
    return key
def eMessage(msg, q, h, g):
    en_msg = []
    k = keygen(q)
    s = powerof(h, k, q)
    p = powerof(g, k, q)
    for i in range(0, len(msg)):
        en_msg.append(msg[i])
    print("g^k used : ", p)
    print("g^ak used : ", s)
    for i in range(0, len(en_msg)):
        en_msg[i] = s * ord(en_msg[i])
    return en_msg, p
def dMessage(en_msg, p, key, q):
    dr_msg = []
    h = powerof(p, key, q)
    for i in range(0, len(en_msg)):
```

```
            dr_msg.append(chr(int(en_msg[i]/h)))
        return dr_msg
    def main():
        print("Enter message:")
        msg=input()
        print("Original Message :", msg)
        q = random.randint(pow(10, 20), pow(10, 50))
        g = random.randint(2, q)
        key = keygen(q)
        h = powerof(g, key, q)
        print("g used : ", g)
        print("g^a used : ", h)
        en_msg, p = eMessage(msg, q, h, g)
        dr_msg = dMessage(en_msg, p, key, q)
        dmsg = ''.join(dr_msg)
        print("Decrypted Message :", dmsg);
    if __name__ == '__main__':
        main()
```

CODE AND OUTPUT SNIPPET





(In the given Spyder Enviroment, the code is mentioned on the left side whereas the output is displayed on the right side)

OUTPUT (from console)

```
runfile('D:/College stuff/Semester 4/Cryptography
Fundamentals/LAB/Assessment 4/Elgamal/elgamal.py', wdir='D:/College
stuff/Semester 4/Cryptography Fundamentals/LAB/Assessment 4/Elgamal')

Enter message:


encryption

Original Message : encryption

g used :  39744912863771025146030351619581775457893422554347

g^a used :  22335954764144156063691409567037709112085744352997

g^k used :  94307241814062013865871247351800042715222722886225

g^ak used :  49510402738303050760013257117136042785687486032681

Decrypted Message : encryption


runfile('D:/College stuff/Semester 4/Cryptography
Fundamentals/LAB/Assessment 4/Elgamal/elgamal.py', wdir='D:/College
stuff/Semester 4/Cryptography Fundamentals/LAB/Assessment 4/Elgamal')

Enter message:


charan
```

```
Original Message : charan

g used :  30504807803800342320573430723194589789390389320333

g^a used :  19991245682566250655938545314190377567318520053805

g^k used :  67571284364176279972854828958070269799228849093 89

g^ak used :  23644018343536555202055342244211934710507473418889

Decrypted Message : charan


runfile('D:/College stuff/Semester 4/Cryptography
Fundamentals/LAB/Assessment 4/Elgamal/elgamal.py', wdir='D:/College
stuff/Semester 4/Cryptography Fundamentals/LAB/Assessment 4/Elgamal')

Enter message:


testing

Original Message : testing

g used :  28291671959606367010080405783455546704224568860366

g^a used :  30668469505374403810860685429532389666037115112521

g^k used :  32080813723713160186589536099793935698698182739736

g^ak used :  32605462231996174626898357882882034280551819461006

Decrypted Message : testing
```