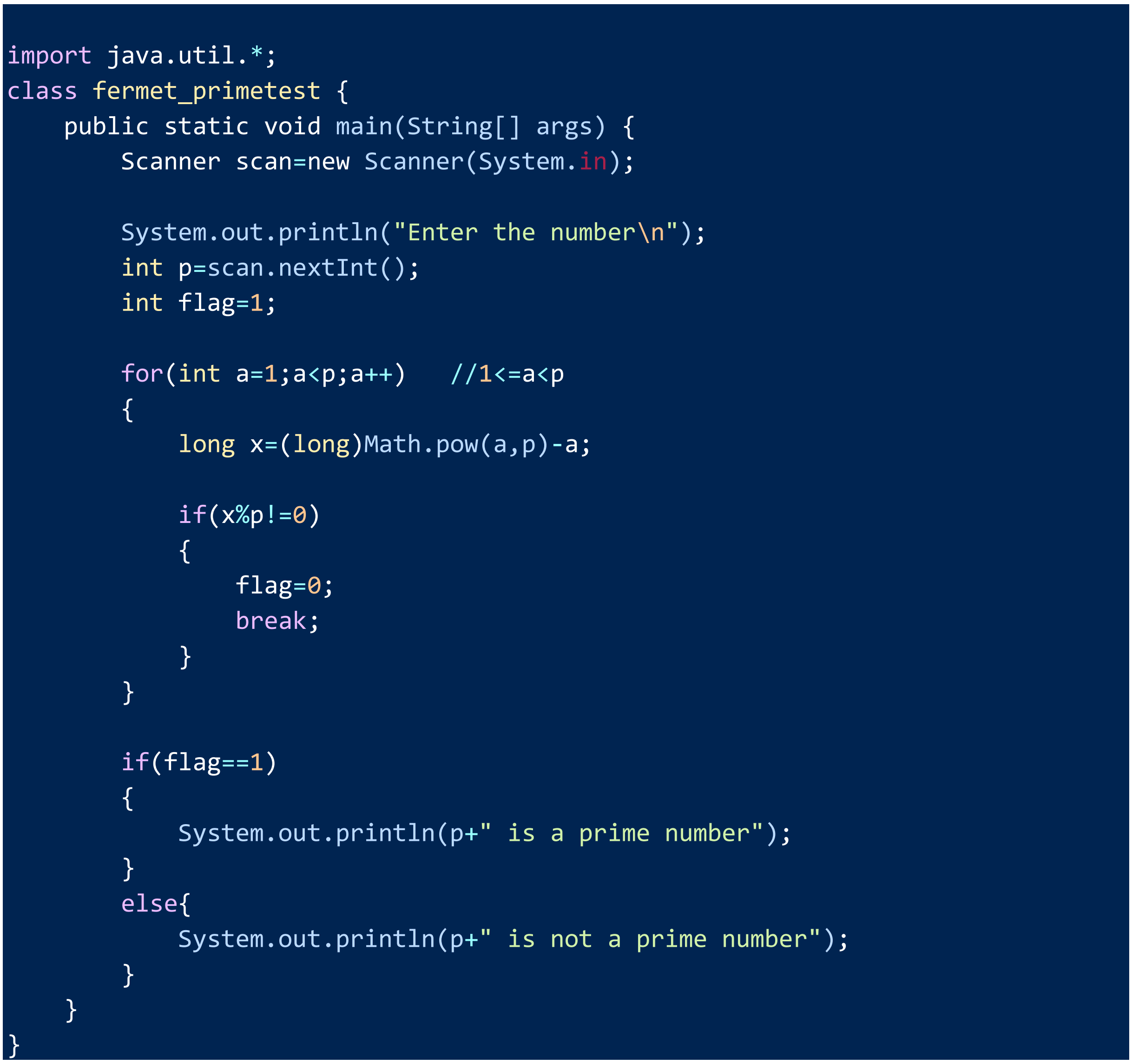
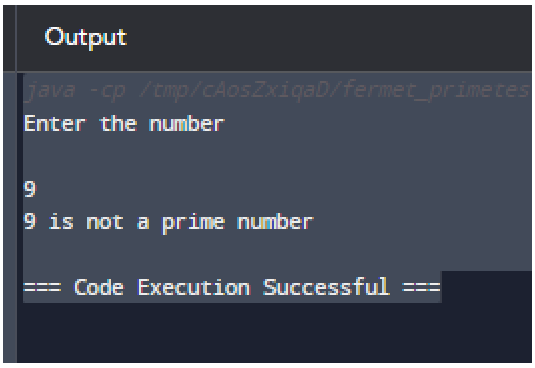
Crypto lab fat

1.Fermet prime_test:

Code:



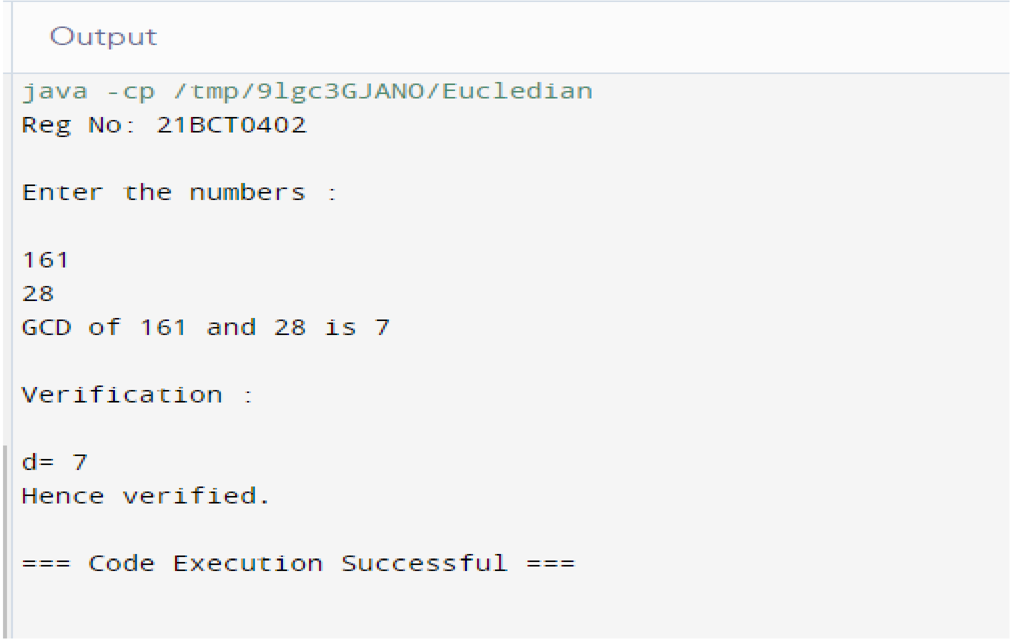
```
import java.util.*;
class fermet_primetest {
    public static void main(String[] args) {
        Scanner scan=new Scanner(System.in);

        System.out.println("Enter the number\n");
        int p=scan.nextInt();
        int flag=1;

        for(int a=1;a<p;a++)    //1<=a<p
        {
            long x=(long)Math.pow(a,p)-a;

            if(x%p!=0)
            {
                flag=0;
                break;
            }
        }

        if(flag==1)
        {
            System.out.println(p+" is a prime number");
        }
        else{
            System.out.println(p+" is not a prime number");
        }
    }
}
```



```
java -cp /tmp/cAosZxiqaD/fermet_primetes
Enter the number

9
9 is not a prime number

=== Code Execution Successful ===
```

2.Gcd using extended eucledian

```
import java.util.*;
public class Eucledian{
 public static void main(String args[])
 {
 Scanner scan=new Scanner(System.in);
 System.out.println("Reg No: 21BCT0402\n");

 System.out.println("Enter the numbers :\n");

 int a=scan.nextInt();
 int b=scan.nextInt();

 //initialization
 int s1=1;
 int s2=0;
 int t1=0;
 int t2=1;

 int r1=a;
 int r2=b;

 int s=0,t=0,q=0,r=0;
 while(r2!=0)
 {
 q=r1/r2;
 r=r1%r2;
 s=s1-s2*q;
 t=t1-t2*q;
 r1=r2;
 r2=r;
 s1=s2;
 s2=s;
 t1=t2;
 t2=t;
 }
 System.out.println("GCD of "+a+" and "+b+" is "+r1);
 System.out.println("\nVerification :\n");
 int d=(a*s1)+(b*t1);
 System.out.println("d= "+d);
 if(d==r1) //d==gcd
 System.out.println("Hence verified.");

 }
}
```

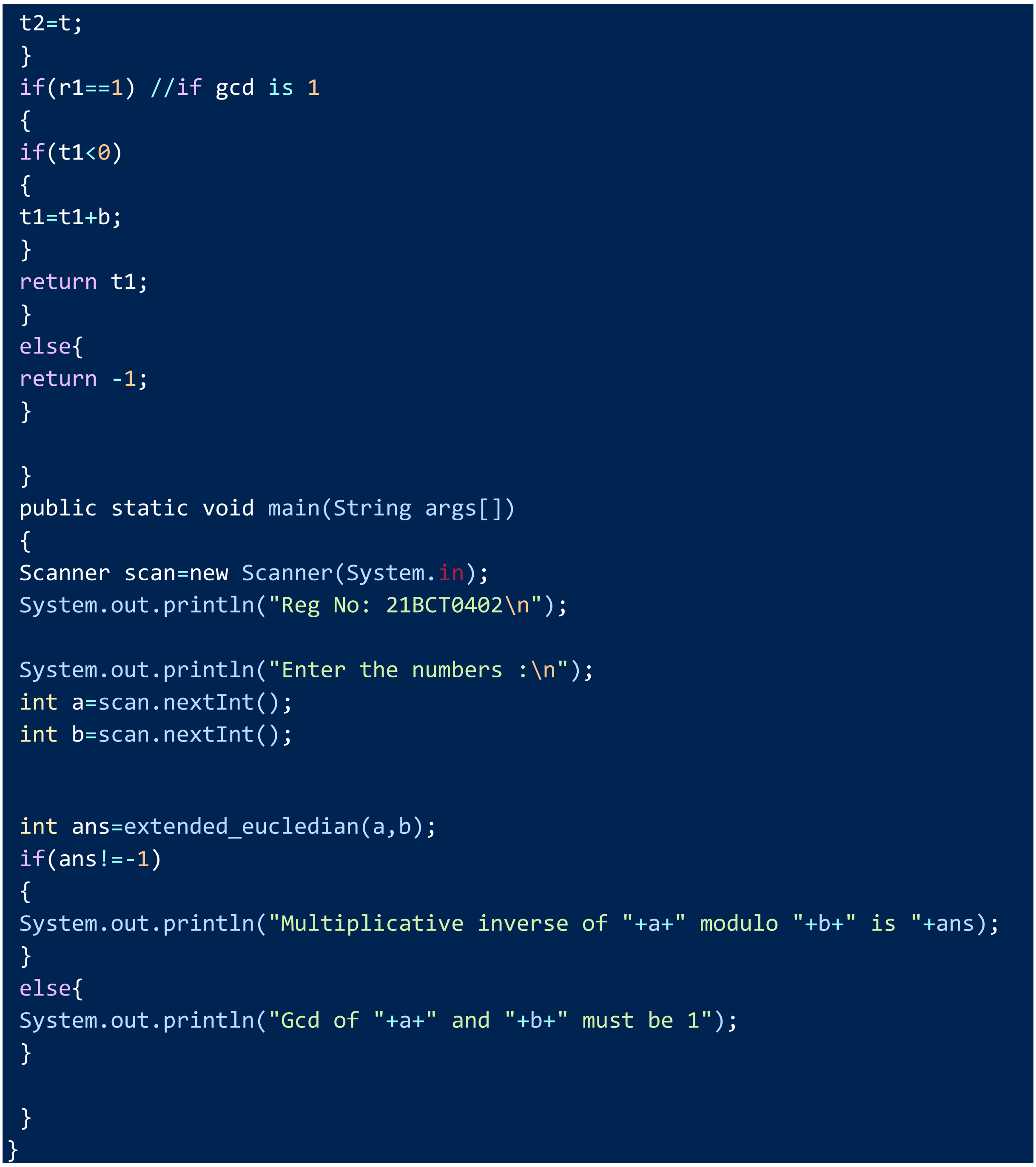


```
Output
java -cp /tmp/9lgc3GJANO/Eucledian
Reg No: 21BCT0402

Enter the numbers :

161
28
GCD of 161 and 28 is 7

Verification :

d= 7
Hence verified.

=== Code Execution Successful ===
```

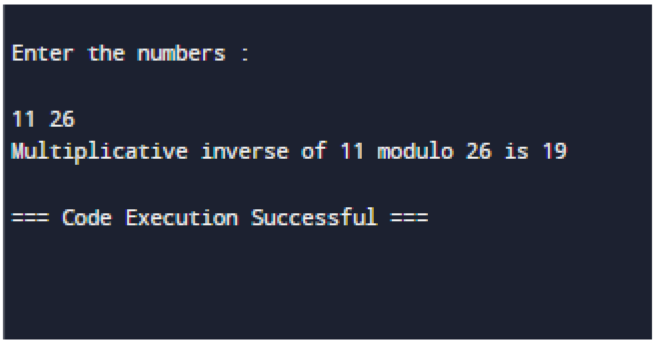
3.Multiplicative inverse using extended eucledian algorithm;

Code:

```
import java.util.*;
public class Multiplicative_inverse{

 public static int extended_eucledian(int a,int b)
 {

 //initialization
 int t1=0;
 int t2=1;

 int r1=b;
 int r2=a;

 int t=0,q=0,r=0;
 while(r2!=0)
 {
 q=r1/r2;
 r=r1%r2;
 t=t1-t2*q;
 r1=r2;
 r2=r;
 t1=t2;
```



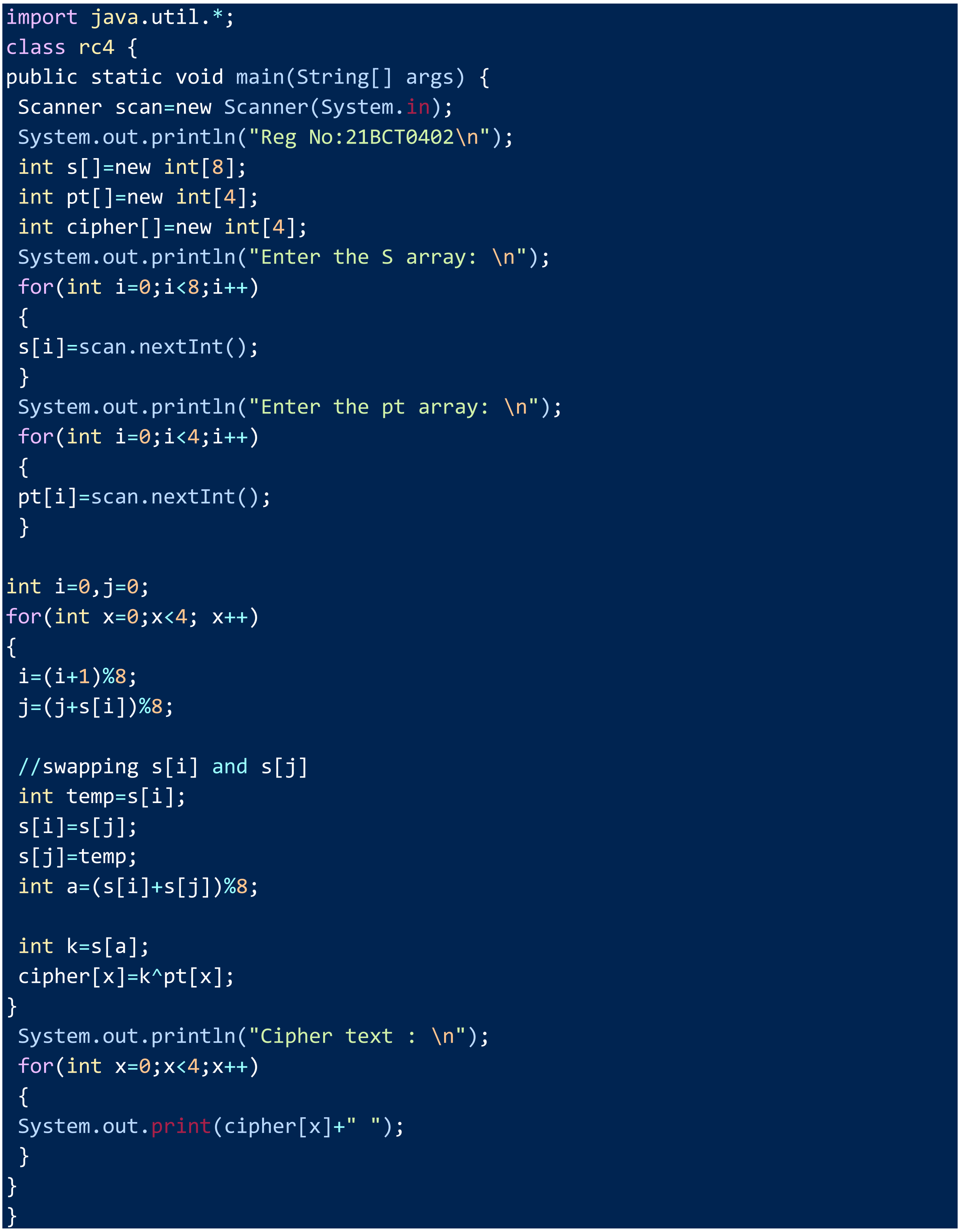
```
 t2=t;
 }
 if(r1==1) //if gcd is 1
 {
 if(t1<0)
 {
 t1=t1+b;
 }
 return t1;
 }
 else{
 return -1;
 }

 }
 public static void main(String args[])
 {
 Scanner scan=new Scanner(System.in);
 System.out.println("Reg No: 21BCT0402\n");

 System.out.println("Enter the numbers :\n");
 int a=scan.nextInt();
 int b=scan.nextInt();


 int ans=extended_eucledian(a,b);
 if(ans!=-1)
 {
 System.out.println("Multiplicative inverse of "+a+" modulo "+b+" is "+ans);
 }
 else{
 System.out.println("Gcd of "+a+" and "+b+" must be 1");
 }

 }
}
```



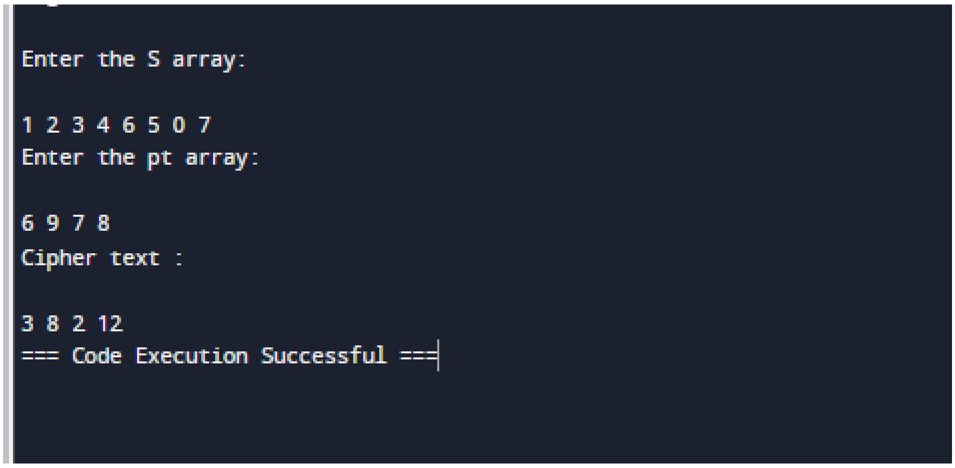
```
Enter the numbers :

11 26
Multiplicative inverse of 11 modulo 26 is 19

=== Code Execution Successful ===
```

4.RC4 ALGORITHM;

Code:



```
import java.util.*;
class rc4 {
public static void main(String[] args) {
 Scanner scan=new Scanner(System.in);
 System.out.println("Reg No:21BCT0402\n");
 int s[]=new int[8];
 int pt[]=new int[4];
 int cipher[]=new int[4];
 System.out.println("Enter the S array: \n");
 for(int i=0;i<8;i++)
 {
 s[i]=scan.nextInt();
 }
 System.out.println("Enter the pt array: \n");
 for(int i=0;i<4;i++)
 {
 pt[i]=scan.nextInt();
 }

int i=0,j=0;
for(int x=0;x<4; x++)
{
 i=(i+1)%8;
 j=(j+s[i])%8;

 //swapping s[i] and s[j]
 int temp=s[i];
 s[i]=s[j];
 s[j]=temp;
 int a=(s[i]+s[j])%8;

 int k=s[a];
 cipher[x]=k^pt[x];
}
 System.out.println("Cipher text : \n");
 for(int x=0;x<4;x++)
 {
 System.out.print(cipher[x]+" ");
 }
}
}
```

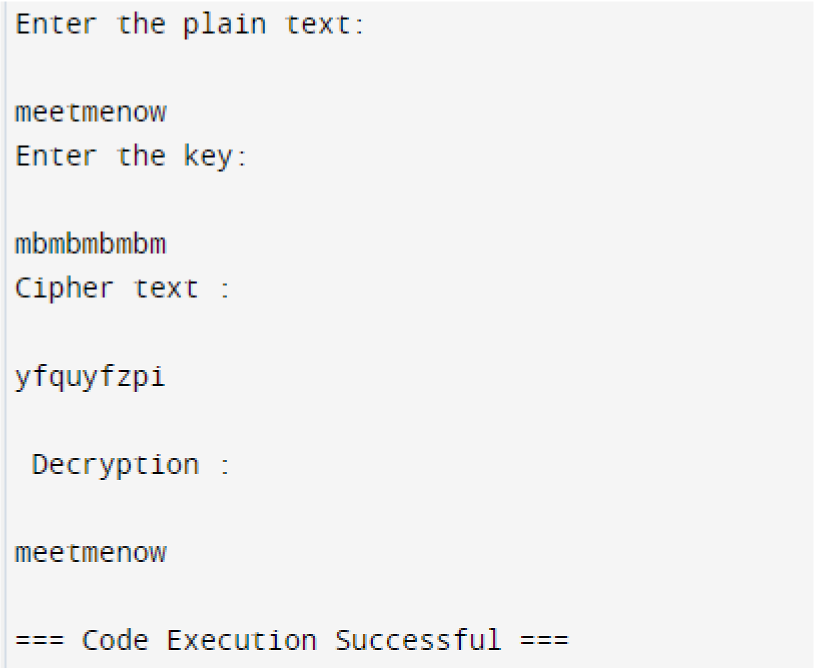


```
Enter the S array:

1 2 3 4 6 5 0 7
Enter the pt array:

6 9 7 8
Cipher text :

3 8 2 12
=== Code Execution Successful ===
```

5. Vernam cipher

Code:

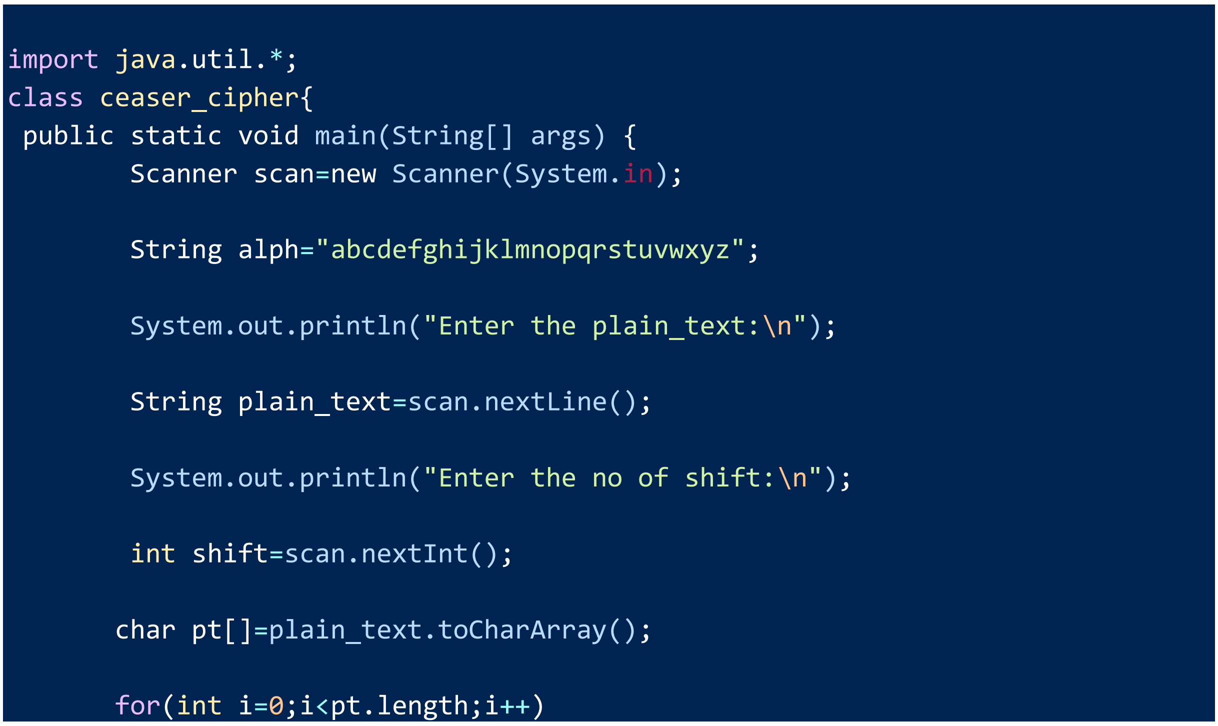
```
import java.util.*;
class Vernam_cipher{
 public static void main(String[] args) {
 Scanner scan=new Scanner(System.in);
 System.out.println("Reg No: 21BCT0402\n");

 System.out.println("Enter the plain text: \n");
 String plain_text=scan.next();

 System.out.println("Enter the key: \n");
 String key=scan.next();

 String alph="abcdefghijklmnopqrstuvwxyz";


 int n=plain_text.length();

 int temp[]=new int[n];

 for(int i=0;i<n;i++)
 {
 int val=alph.indexOf(plain_text.charAt(i))+alph.indexOf(key.charAt(i));
 if(val>25)
 temp[i]=val-26;
 else
 temp[i]=val;

 }

 //Cipher text
 String ct="";
 System.out.println("Cipher text : \n");
 for(int i=0;i<n;i++)
 {
 ct+=alph.charAt(temp[i]);
 }
 System.out.println(ct);

 //Decryption
 String ctd="";
 int temp2[]=new int[n];
 for(int i=0;i<n;i++)
 {
 int val=alph.indexOf(ct.charAt(i))-alph.indexOf(key.charAt(i));

 if(val<0)
 temp2[i]=val+26;
 else
 temp2[i]=val;

 }
 System.out.println("\n Decryption :\n");
 for(int i=0;i<n;i++)
 {
 ctd+=alph.charAt(temp2[i]);
 }
 System.out.println(ctd);

 }
}
```



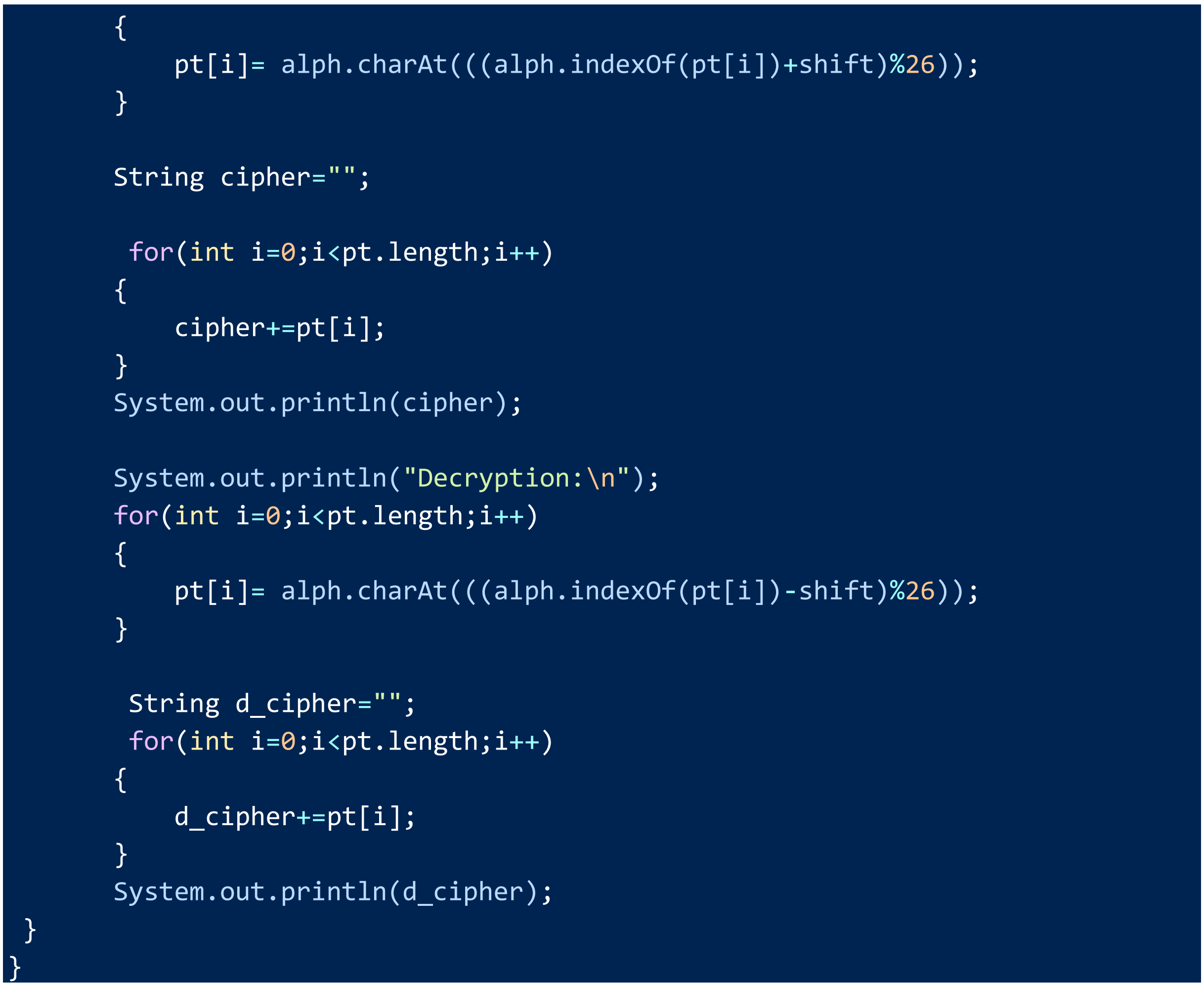
```
Enter the plain text:

meetmenow
Enter the key:

mbmbmbmbm
Cipher text :

yfquyfzpi

 Decryption :

meetmenow

=== Code Execution Successful ===
```

6.Ceaser cipher:

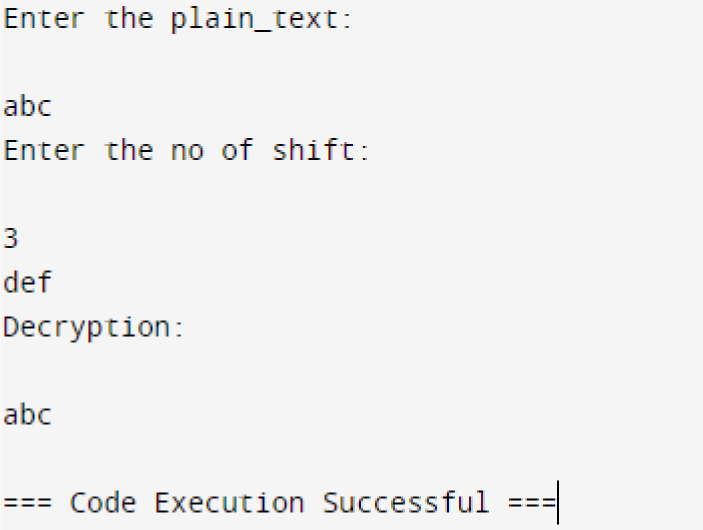
Code:



```
import java.util.*;
class ceaser_cipher{
 public static void main(String[] args) {
        Scanner scan=new Scanner(System.in);

        String alph="abcdefghijklmnopqrstuvwxyz";

        System.out.println("Enter the plain_text:\n");

        String plain_text=scan.nextLine();

        System.out.println("Enter the no of shift:\n");

        int shift=scan.nextInt();

       char pt[]=plain_text.toCharArray();

       for(int i=0;i<pt.length;i++)
```



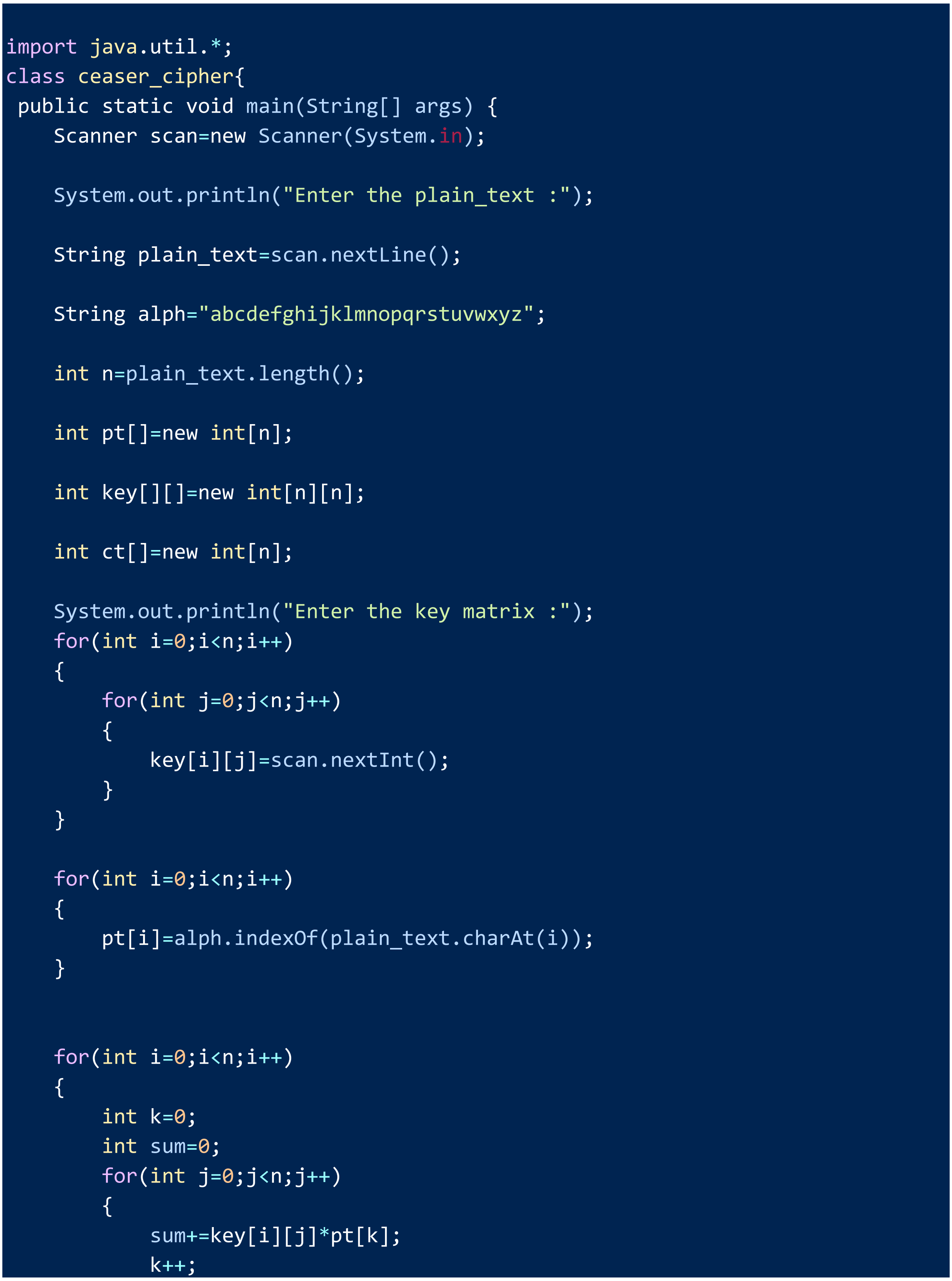
```
       {
           pt[i]= alph.charAt(((alph.indexOf(pt[i])+shift)%26));
       }

       String cipher="";

        for(int i=0;i<pt.length;i++)
       {
           cipher+=pt[i];
       }
       System.out.println(cipher);

       System.out.println("Decryption:\n");
       for(int i=0;i<pt.length;i++)
       {
           pt[i]= alph.charAt(((alph.indexOf(pt[i])-shift)%26));
       }

        String d_cipher="";
        for(int i=0;i<pt.length;i++)
       {
           d_cipher+=pt[i];
       }
       System.out.println(d_cipher);
 }
}
```



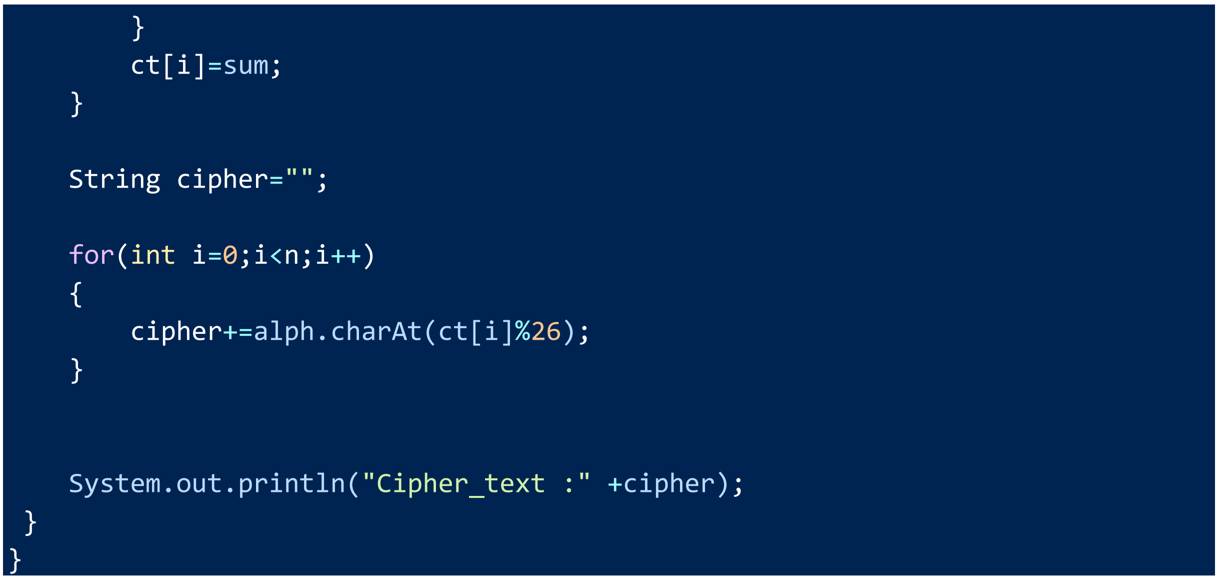
```
Enter the plain_text:

abc
Enter the no of shift:

3
def
Decryption:

abc

=== Code Execution Successful ===
```

7.Hill cipher:

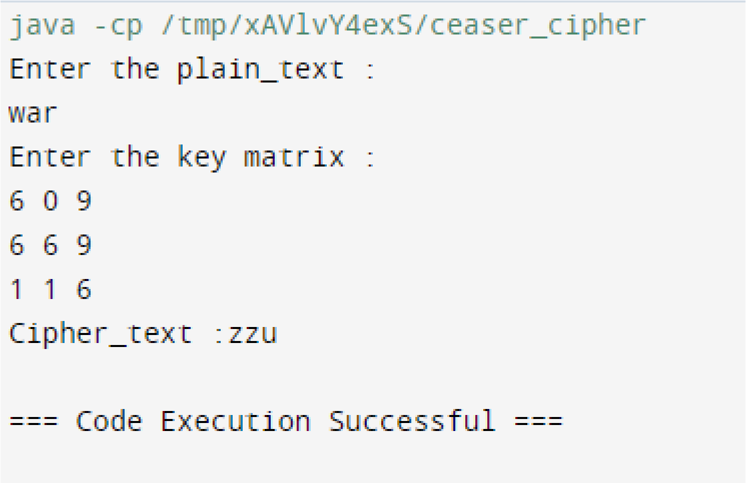
Code:



```
import java.util.*;
class ceaser_cipher{
 public static void main(String[] args) {
    Scanner scan=new Scanner(System.in);

    System.out.println("Enter the plain_text :");

    String plain_text=scan.nextLine();

    String alph="abcdefghijklmnopqrstuvwxyz";

    int n=plain_text.length();

    int pt[]=new int[n];

    int key[][]=new int[n][n];

    int ct[]=new int[n];

    System.out.println("Enter the key matrix :");
    for(int i=0;i<n;i++)
    {
        for(int j=0;j<n;j++)
        {
            key[i][j]=scan.nextInt();
        }
    }

    for(int i=0;i<n;i++)
    {
        pt[i]=alph.indexOf(plain_text.charAt(i));
    }


    for(int i=0;i<n;i++)
    {
        int k=0;
        int sum=0;
        for(int j=0;j<n;j++)
        {
            sum+=key[i][j]*pt[k];
            k++;
```



```
        }
        ct[i]=sum;
    }

    String cipher="";

    for(int i=0;i<n;i++)
    {
        cipher+=alph.charAt(ct[i]%26);
    }


    System.out.println("Cipher_text :" +cipher);
 }
}
```

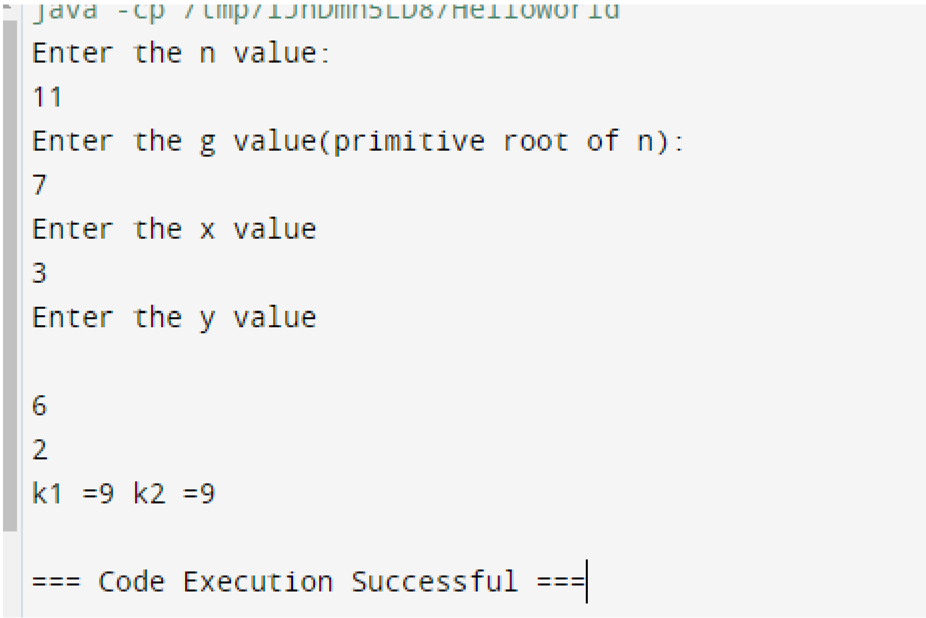


```
java -cp /tmp/xAVlvY4exS/ceaser_cipher
Enter the plain_text :
war
Enter the key matrix :
6 0 9
6 6 9
1 1 6
Cipher_text :zzu

=== Code Execution Successful ===
```

8.Diffie helman key exchange:

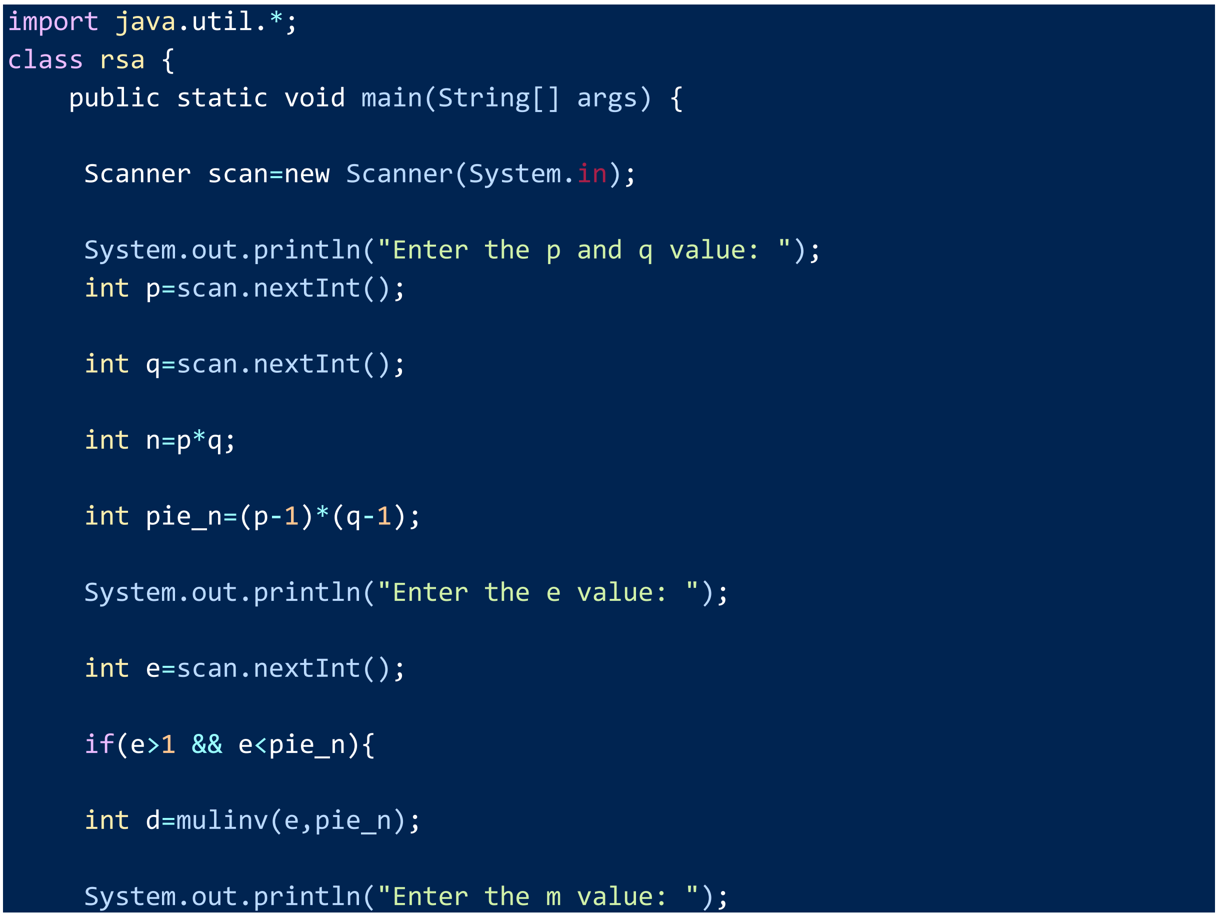
Code:

```
import java.util.*;
class Diffie_helman {
    public static void main(String[] args) {
        Scanner scan=new Scanner(System.in);
        System.out.println("Enter the n value:");
        int n=scan.nextInt();

        System.out.println("Enter the g value(primitive root of n):");
        int g=scan.nextInt();

        if(g<n){

        System.out.println("Enter the x value");
        int x=scan.nextInt();

        System.out.println("Enter the y value");
        int y=scan.nextInt();

        int A=(int)Math.pow(g,x)%n;


        int B=(int)Math.pow(g,y)%n;

        int k1=(int)Math.pow(B,x)%n;

        int k2=(int)Math.pow(A,y)%n;

         System.out.println("k1 ="+k1+" k2 ="+k2);

        }
        else{
            System.out.println("g must be less than n");
        }
    }
}
```

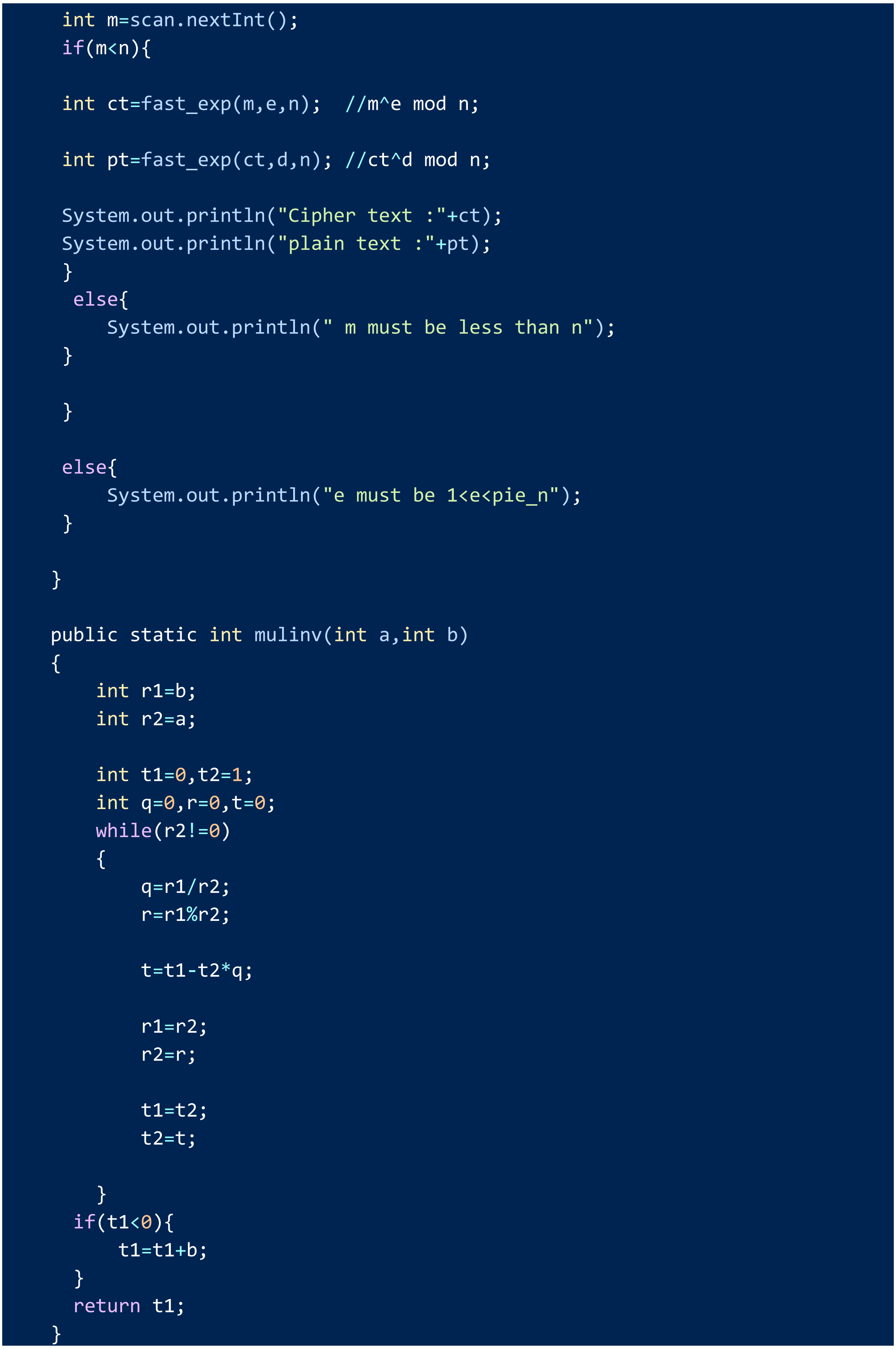


```
java -cp /tmp/1JnDmn5LD8/HelloWorld
Enter the n value:
11
Enter the g value(primitive root of n):
7
Enter the x value
3
Enter the y value

6
2
k1 =9 k2 =9

=== Code Execution Successful ===
```

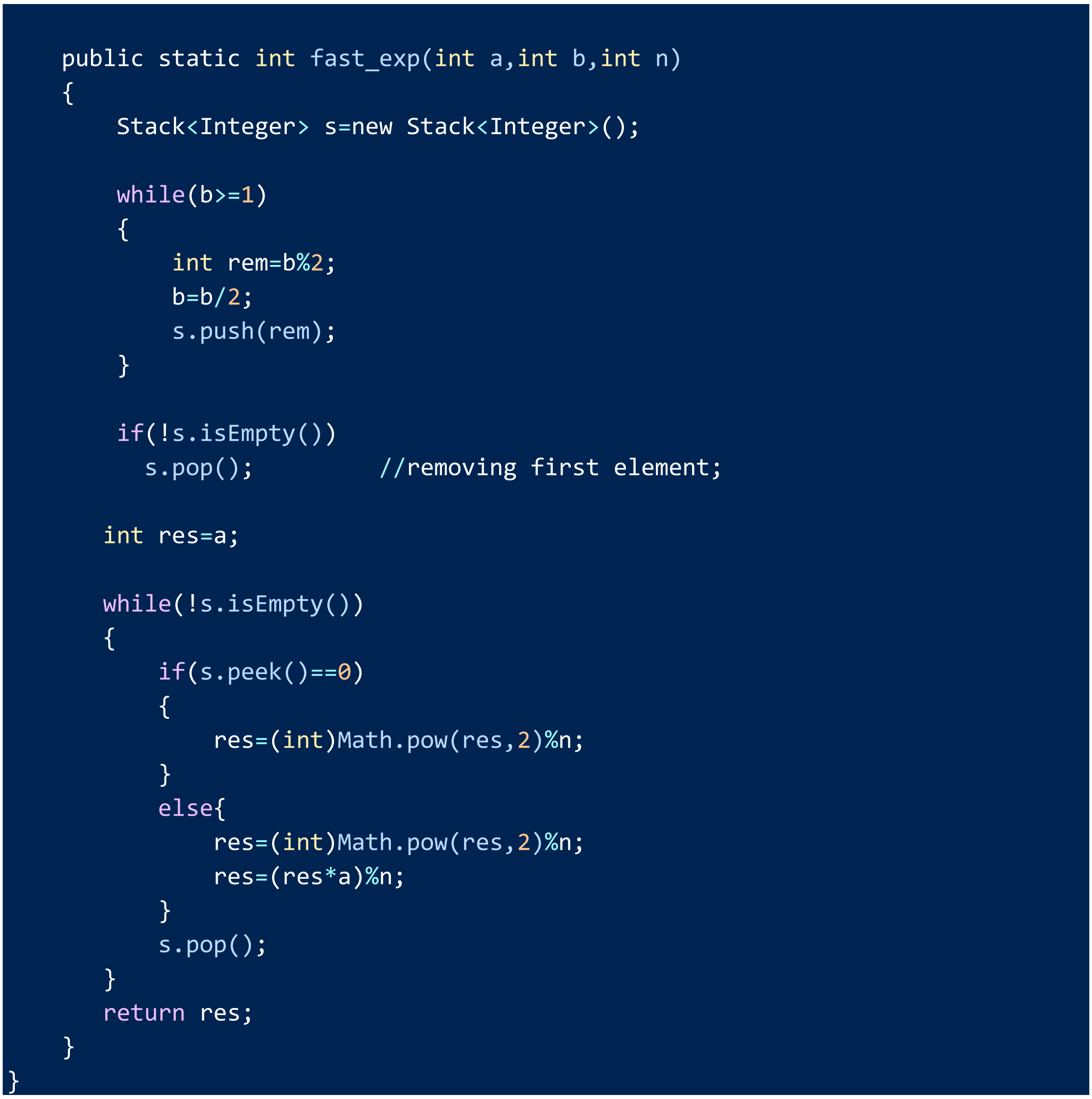
9.RSA Algorithm:



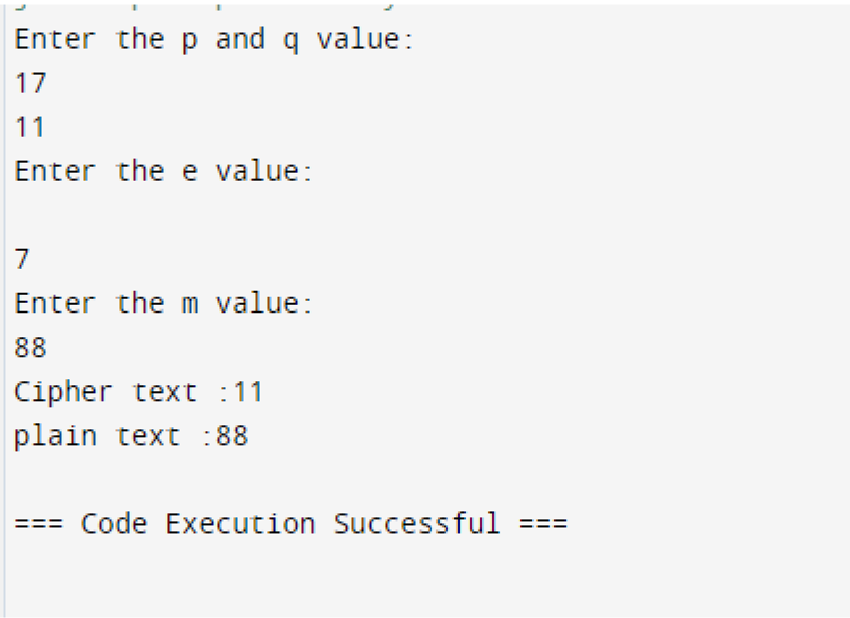
```
import java.util.*;
class rsa {
    public static void main(String[] args) {

     Scanner scan=new Scanner(System.in);

     System.out.println("Enter the p and q value: ");
     int p=scan.nextInt();

     int q=scan.nextInt();

     int n=p*q;

     int pie_n=(p-1)*(q-1);

     System.out.println("Enter the e value: ");

     int e=scan.nextInt();

     if(e>1 && e<pie_n){

     int d=mulinv(e,pie_n);

     System.out.println("Enter the m value: ");
```



```
     int m=scan.nextInt();
     if(m<n){

     int ct=fast_exp(m,e,n);  //m^e mod n;

     int pt=fast_exp(ct,d,n); //ct^d mod n;

     System.out.println("Cipher text :"+ct);
     System.out.println("plain text :"+pt);
     }
      else{
         System.out.println(" m must be less than n");
     }

     }

     else{
         System.out.println("e must be 1<e<pie_n");
     }

    }

    public static int mulinv(int a,int b)
    {
        int r1=b;
        int r2=a;

        int t1=0,t2=1;
        int q=0,r=0,t=0;
        while(r2!=0)
        {
            q=r1/r2;
            r=r1%r2;

            t=t1-t2*q;

            r1=r2;
            r2=r;

            t1=t2;
            t2=t;

        }
      if(t1<0){
          t1=t1+b;
      }
      return t1;
    }
```



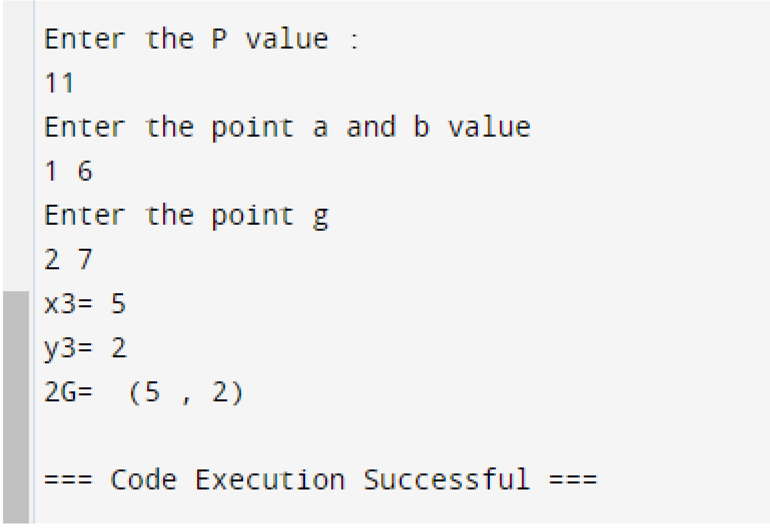
```
    public static int fast_exp(int a,int b,int n)
    {
        Stack<Integer> s=new Stack<Integer>();

        while(b>=1)
        {
            int rem=b%2;
            b=b/2;
            s.push(rem);
        }

        if(!s.isEmpty())
          s.pop();         //removing first element;

       int res=a;

       while(!s.isEmpty())
       {
           if(s.peek()==0)
           {
               res=(int)Math.pow(res,2)%n;
           }
           else{
               res=(int)Math.pow(res,2)%n;
               res=(res*a)%n;
           }
           s.pop();
       }
       return res;
    }
}
```



```
Enter the p and q value:
17
11
Enter the e value:

7
Enter the m value:
88
Cipher text :11
plain text :88

=== Code Execution Successful ===
```

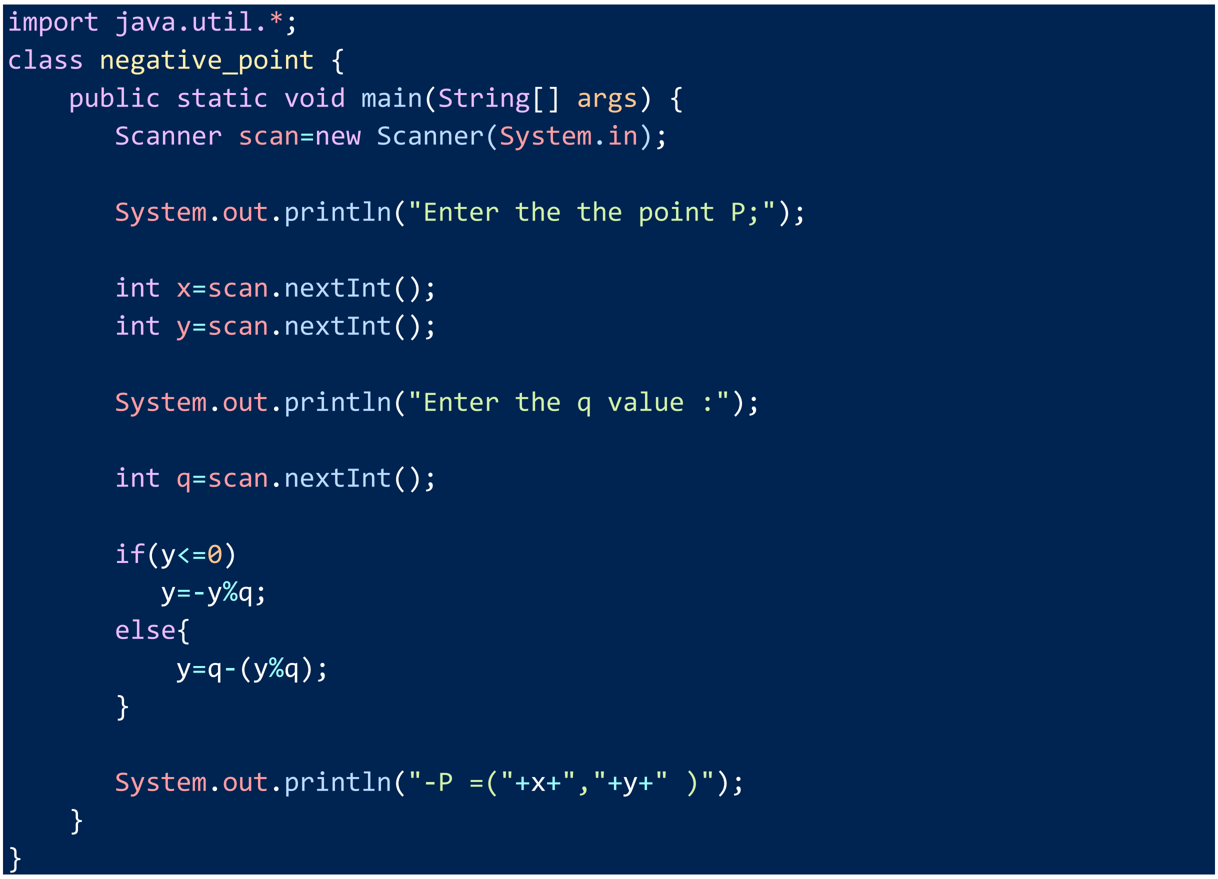
10.point doubling ecc:

```
import java.util.*;
class pointdoubling {
public static void main(String[] args) {
Scanner scan=new Scanner(System.in);
System.out.println("Reg No:21BCT0402");
System.out.println("\nEnter the P value :");
int p=scan.nextInt();
System.out.println("Enter the point a and b value");
int a=scan.nextInt();
int b=scan.nextInt();
System.out.println("Enter the point g");
int x1=scan.nextInt();
int y1=scan.nextInt();

int x2=x1;
int y2=y1;

int inv=multiplicative_inverse((2*y1),p);

int lambda=((3*(int)Math.pow(x1,2)+a)*inv)%p;

int x3=((int)Math.pow(lambda,2)-x1-x2);
if(x3<0)
  x3=p-((x3*-1) %p);
 else{
     x3=x3%p;
 }

int y3=(lambda*(x1-x3)-y1);
if(y3<0)
  y3=p-((y3*-1) %p);
 else{
     y3=y3%p;
 }


System.out.println("x3= "+x3);
System.out.println("y3= "+y3);
System.out.println("2G=  ("+x3 +" , "+y3+")");
}
public static int multiplicative_inverse(int a,int b)
{
//initialization
int t1=0;
int t2=1;
int r1=b;
int r2=a;
int t=0,q=0,r=0;
while(r2!=0)
{
q=r1/r2;
r=r1%r2;
t=t1-t2*q;
r1=r2;
r2=r;
t1=t2;
t2=t;
}

if(t1<0)
{
t1=t1+b;
}
return t1;

}

}
```

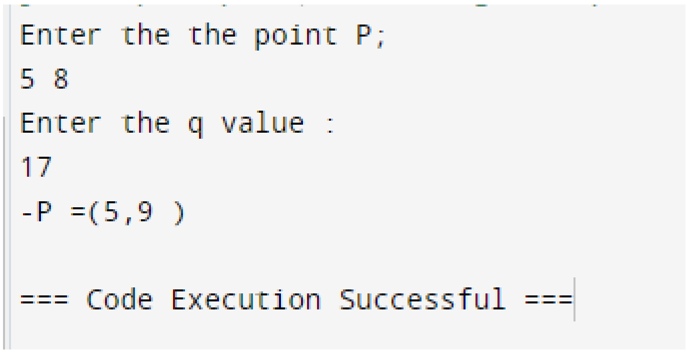


```
Enter the P value :
11
Enter the point a and b value
1 6
Enter the point g
2 7
x3= 5
y3= 2
2G=  (5 , 2)

=== Code Execution Successful ===
```

11.Negative point:



```
import java.util.*;
class negative_point {
    public static void main(String[] args) {
       Scanner scan=new Scanner(System.in);

       System.out.println("Enter the the point P;");

       int x=scan.nextInt();
       int y=scan.nextInt();

       System.out.println("Enter the q value :");

       int q=scan.nextInt();

       if(y<=0)
          y=-y%q;
       else{
           y=q-(y%q);
       }

       System.out.println("-P =("+x+","+y+" )");
    }
}
```

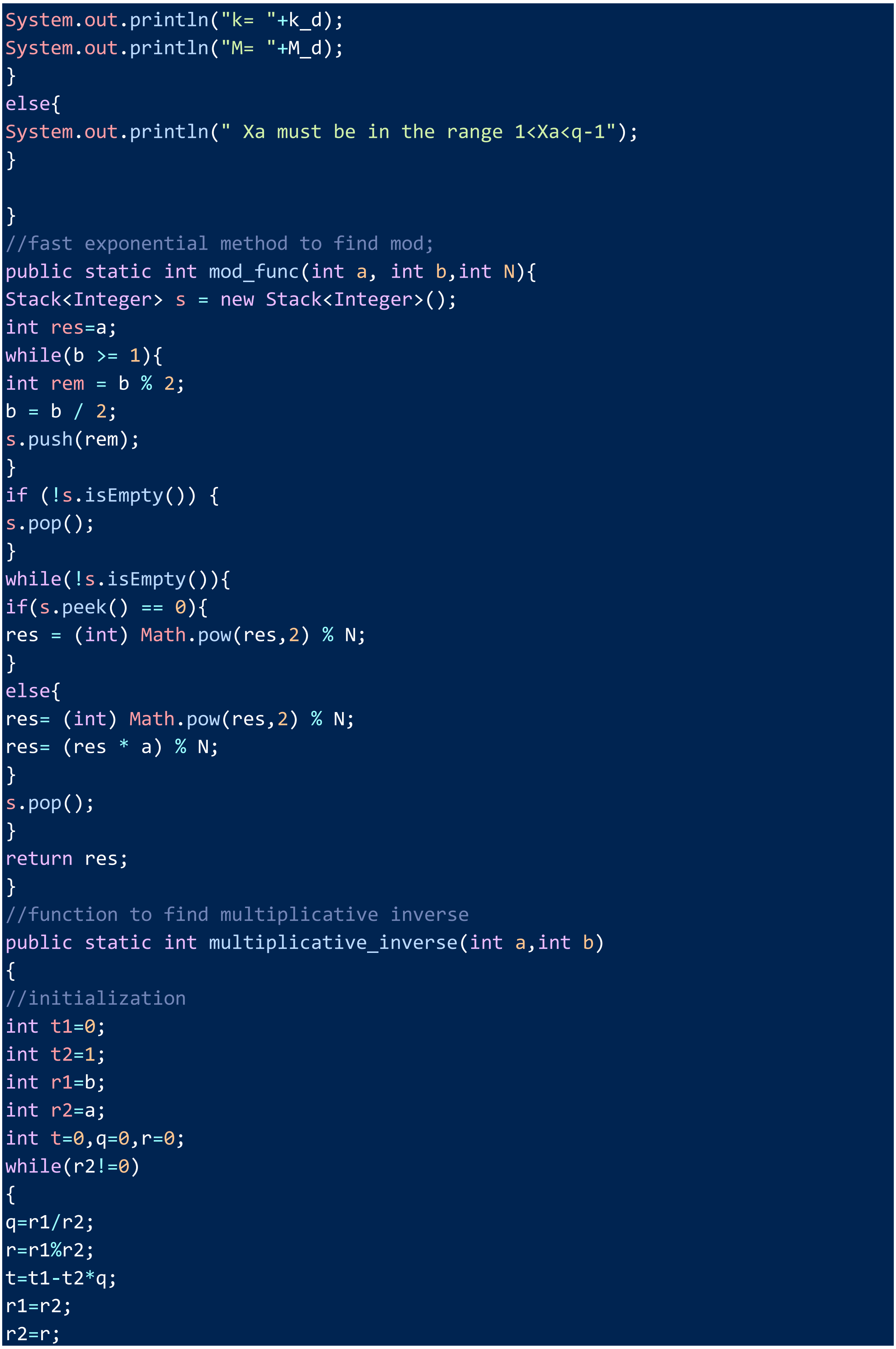


```
Enter the the point P;
5 8
Enter the q value :
17
-P =(5,9 )

=== Code Execution Successful ===
```

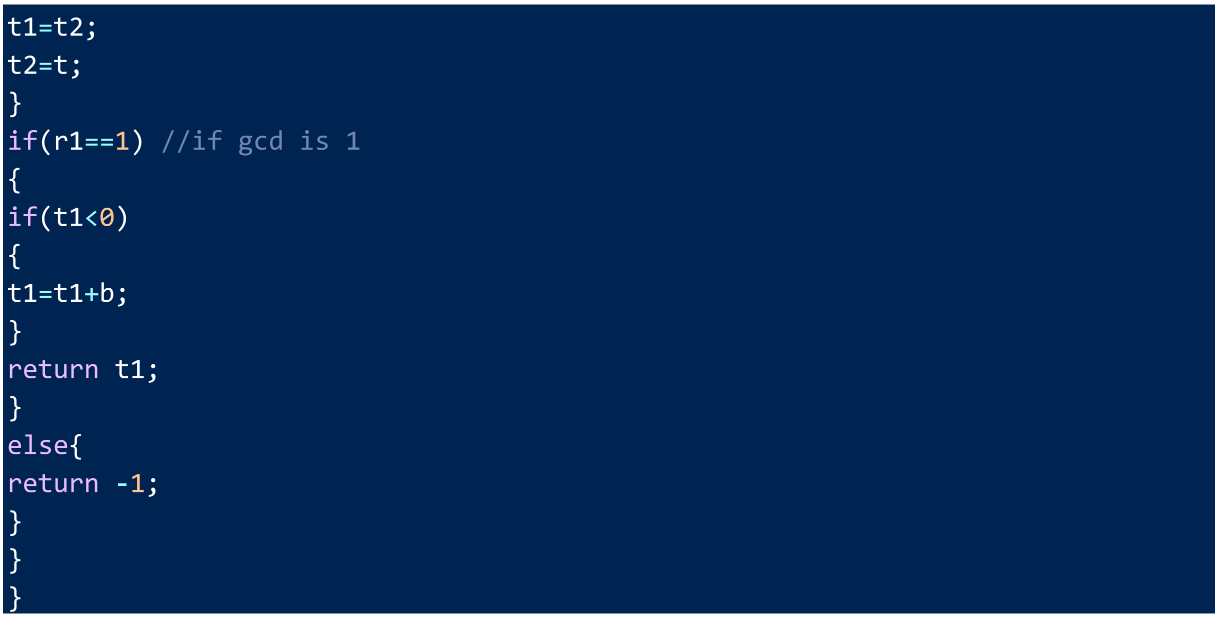
12.Elgamal Cryptography:

Code:

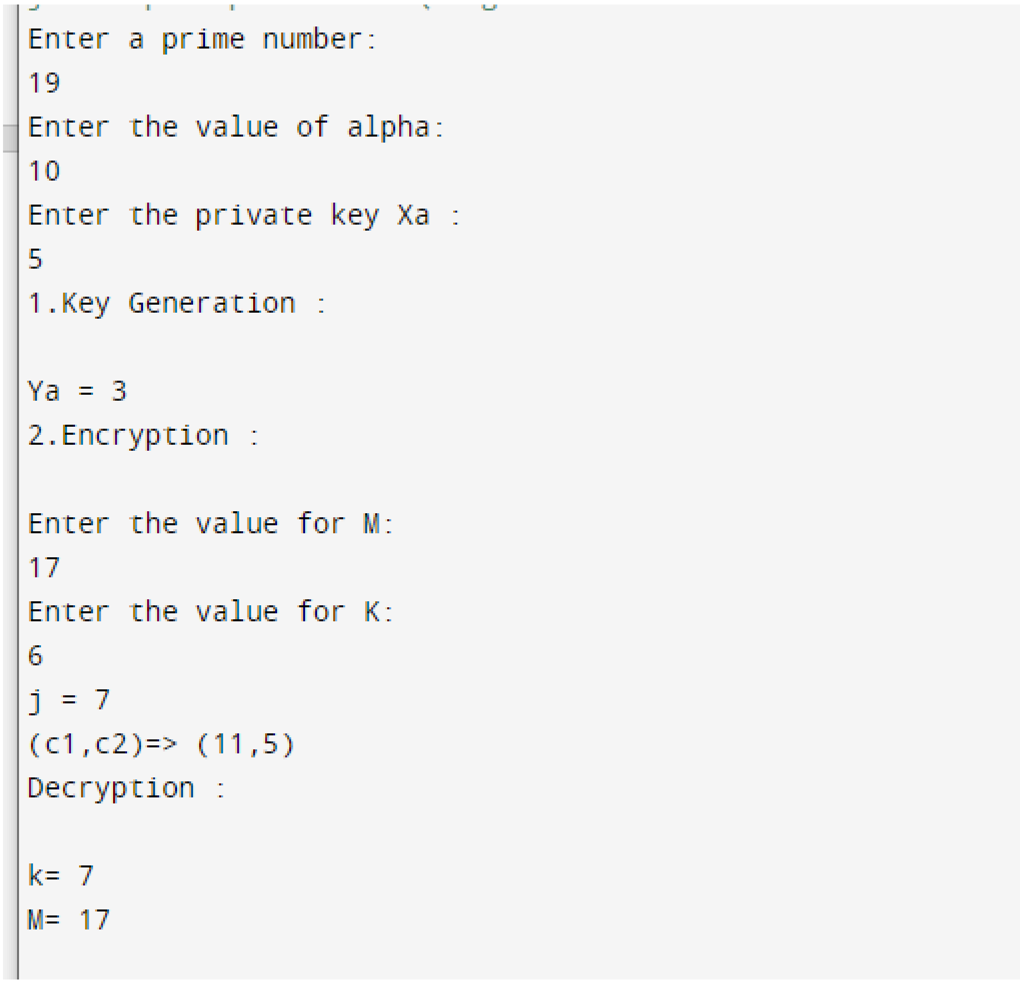
```
import java.util.*;
class Elgamal{
public static void main(String[] args) {
Scanner scan=new Scanner(System.in);
System.out.println("Reg No:21BCT0402");
System.out.println("Enter a prime number:");
int q=scan.nextInt();
System.out.println("Enter the value of alpha:");
int alpha=scan.nextInt();

System.out.println("Enter the private key Xa :");
int Xa=scan.nextInt();
if(Xa>1 && Xa<q-1)
{
int Ya=mod_func(alpha,Xa,q);
System.out.println("1.Key Generation :\n");
System.out.println("Ya = "+Ya);
System.out.println("2.Encryption :\n");
System.out.println("Enter the value for M:");
int M=scan.nextInt();
System.out.println("Enter the value for K:");
int K=scan.nextInt();
int k=mod_func(Ya,K,q);
System.out.println("k = "+k);
int c1=mod_func(alpha,K,q);
int c2=(k*M)%q;
System.out.println("(c1,c2)=> ("+c1+","+c2+")");
System.out.println("Decryption :\n");
int k_d=mod_func(c1,Xa,q);
int inv=multiplicative_inverse(k,q);
int M_d=c2*inv %q;
```



```
System.out.println("k= "+k_d);
System.out.println("M= "+M_d);
}
else{
System.out.println(" Xa must be in the range 1<Xa<q-1");
}

}
//fast exponential method to find mod;
public static int mod_func(int a, int b,int N){
Stack<Integer> s = new Stack<Integer>();
int res=a;
while(b >= 1){
int rem = b % 2;
b = b / 2;
s.push(rem);
}
if (!s.isEmpty()) {
s.pop();
}
while(!s.isEmpty()){
if(s.peek() == 0){
res = (int) Math.pow(res,2) % N;
}
else{
res= (int) Math.pow(res,2) % N;
res= (res * a) % N;
}
s.pop();
}
return res;
}
//function to find multiplicative inverse
public static int multiplicative_inverse(int a,int b)
{
//initialization
int t1=0;
int t2=1;
int r1=b;
int r2=a;
int t=0,q=0,r=0;
while(r2!=0)
{
q=r1/r2;
r=r1%r2;
t=t1-t2*q;
r1=r2;
r2=r;
```



```
t1=t2;
t2=t;
}
if(r1==1) //if gcd is 1
{
if(t1<0)
{
t1=t1+b;
}
return t1;
}
else{
return -1;
}
}
}
```

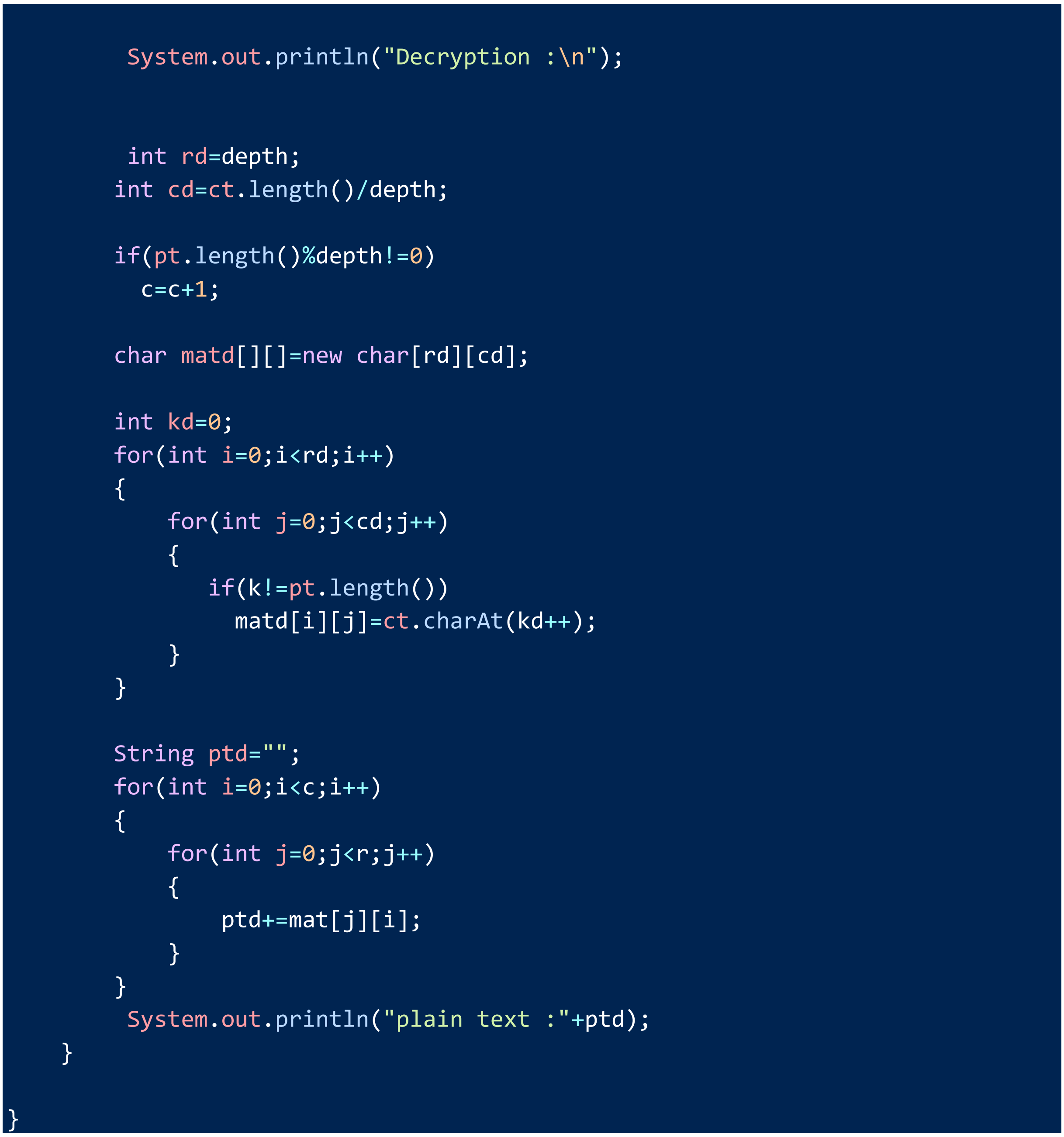


```
Enter a prime number:
19
Enter the value of alpha:
10
Enter the private key Xa :
5
1.Key Generation :

Ya = 3
2.Encryption :

Enter the value for M:
17
Enter the value for K:
6
j = 7
(c1,c2)=> (11,5)
Decryption :

k= 7
M= 17
```

13.RailFence:

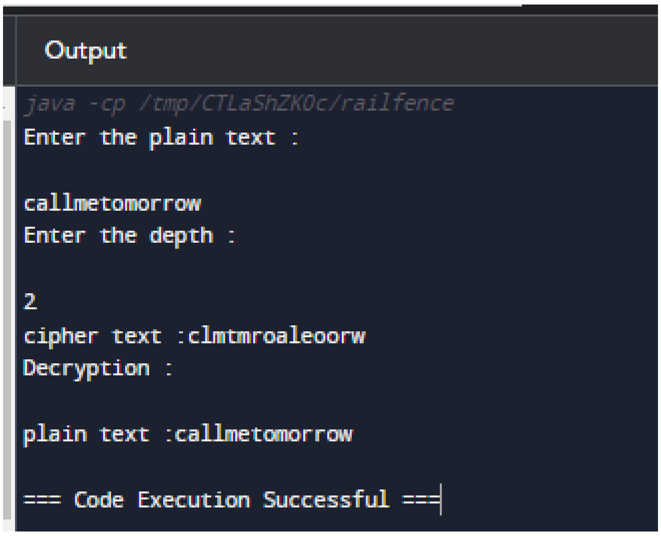
Code:

```
import java.util.*;

class railfence{
    public static void main(String args[])
    {
        Scanner scan=new Scanner(System.in);

        System.out.println("Enter the plain text :\n");

        String pt=scan.nextLine();

        System.out.println("Enter the depth :\n");

        int depth=scan.nextInt();

        int r=depth;
        int c=pt.length()/depth;

        if(pt.length()%depth!=0)
          c=c+1;

        char mat[][]=new char[r][c];

        int k=0;
        for(int i=0;i<c;i++)
        {
            for(int j=0;j<r;j++)
            {
               if(k!=pt.length())
                 mat[j][i]=pt.charAt(k++);
            }
        }

        String ct="";
        for(int i=0;i<r;i++)
        {
            for(int j=0;j<c;j++)
            {
                ct+=mat[i][j];
            }
        }

        System.out.println("cipher text :"+ct);
```



```
         System.out.println("Decryption :\n");


         int rd=depth;
        int cd=ct.length()/depth;

        if(pt.length()%depth!=0)
          c=c+1;

        char matd[][]=new char[rd][cd];

        int kd=0;
        for(int i=0;i<rd;i++)
        {
            for(int j=0;j<cd;j++)
            {
               if(k!=pt.length())
                 matd[i][j]=ct.charAt(kd++);
            }
        }

        String ptd="";
        for(int i=0;i<c;i++)
        {
            for(int j=0;j<r;j++)
            {
                ptd+=mat[j][i];
            }
        }
         System.out.println("plain text :"+ptd);
    }

}
```

Output:

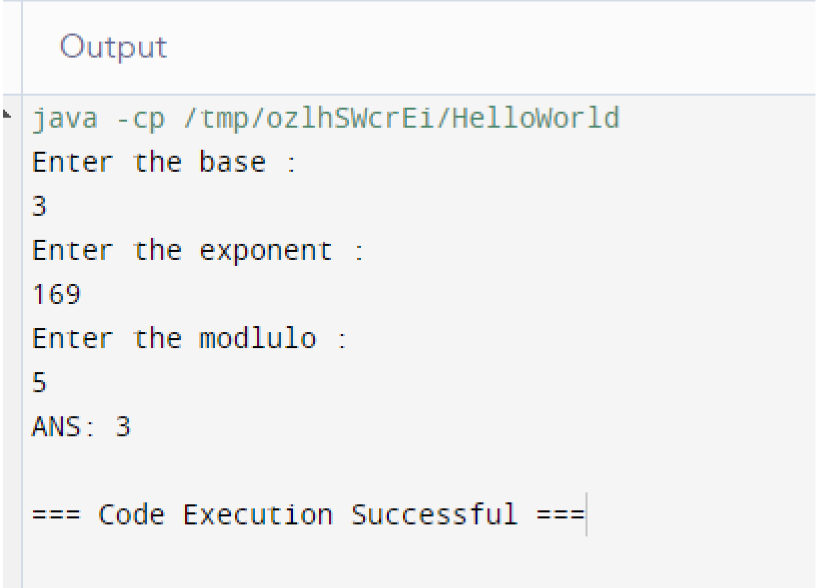


```
java -cp /tmp/CTLaShZK0c/railfence
Enter the plain text :

callmetomorrow
Enter the depth :

2
cipher text :clmtmroaleoorw
Decryption :

plain text :callmetomorrow

=== Code Execution Successful ===
```

14.Eulers theorem("3 cases"):

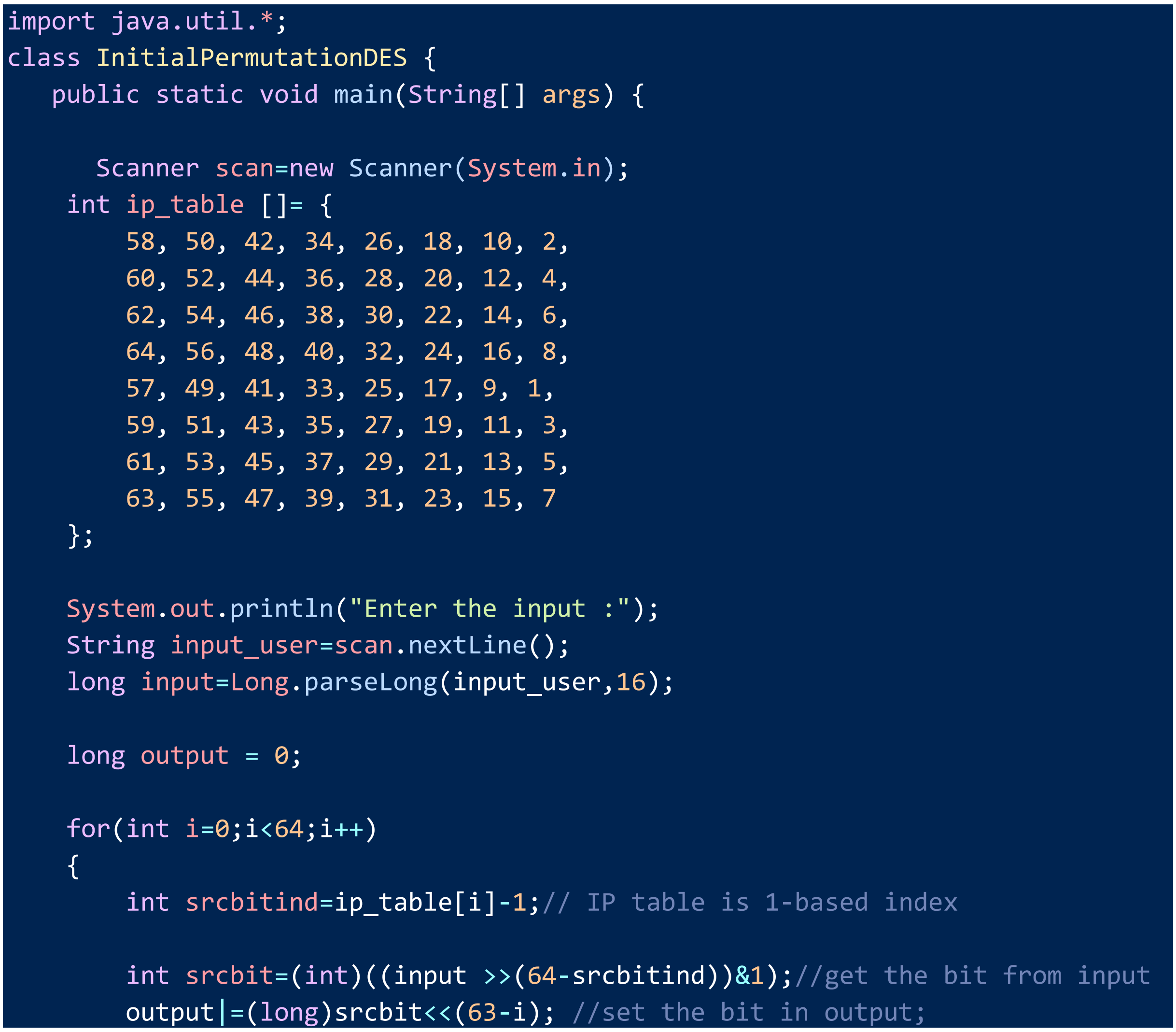
Code:

```
import java.util.*;

class HelloWorld {
    public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);

        System.out.println("Enter the base :");
        int a = scan.nextInt();
        System.out.println("Enter the exponent :");
        int b = scan.nextInt();

        System.out.println("Enter the modlulo :");
        int n = scan.nextInt();


        int pie_n = 0;
        int p = 0, q = 0,e=0;

        //CASE 1
        if (isPrime(n)) {
            pie_n = n - 1;
        }

        //CASE 3
         else if (!isPrime(n) ) {
                for (int i = 1; i < n; i++) {
                    for (int j = 1; j < n; j++) {
                        if ((int)Math.pow(i, j) == n) {
                            p= i;
                            e = j;
                        }
                    }
                }
                pie_n = (int)Math.pow(p,e) - (int)Math.pow(p,  e- 1);
            }
            else if(p==0 && e==0) {
          //CASE2
            for (int i = 1; i < n; i++) {
                for (int j = 1; j < n; j++) {
                    if (i * j == n) {
                        if (isPrime(i) && isPrime(j)) {
                            p = i;
                            q = j;
                        }

                    }
                }
            }
            pie_n = (p - 1) * (q - 1);

        }


        int pow = b % pie_n;

        int ans =1*(int)Math.pow(a, pow) % n;

        System.out.println("ANS: "+ans);
    }

    public static boolean isPrime(int n) {
        if (n <= 1) {
            return false;
        }
        for (int i = 2; i < n; i++) {
            if (n % i == 0) {
                return false;
            }
        }
        return true;
    }
}
```



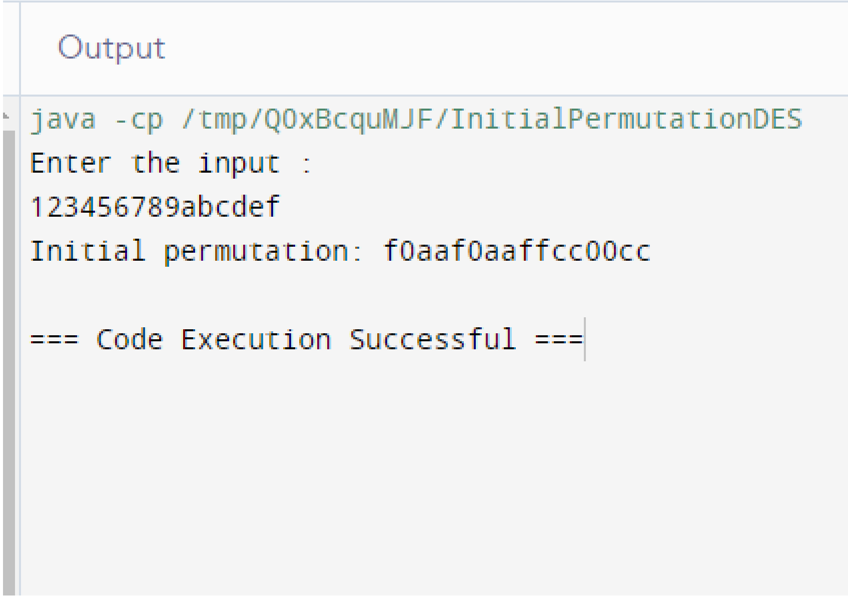
```
java -cp /tmp/ozlhSWcrEi/HelloWorld
Enter the base :
3
Enter the exponent :
169
Enter the modlulo :
5
ANS: 3

=== Code Execution Successful ===
```

15.Initial permutation in des:



```
import java.util.*;
class InitialPermutationDES {
   public static void main(String[] args) {

      Scanner scan=new Scanner(System.in);
    int ip_table []= {
        58, 50, 42, 34, 26, 18, 10, 2,
        60, 52, 44, 36, 28, 20, 12, 4,
        62, 54, 46, 38, 30, 22, 14, 6,
        64, 56, 48, 40, 32, 24, 16, 8,
        57, 49, 41, 33, 25, 17, 9, 1,
        59, 51, 43, 35, 27, 19, 11, 3,
        61, 53, 45, 37, 29, 21, 13, 5,
        63, 55, 47, 39, 31, 23, 15, 7
    };

    System.out.println("Enter the input :");
    String input_user=scan.nextLine();
    long input=Long.parseLong(input_user,16);

    long output = 0;

    for(int i=0;i<64;i++)
    {
        int srcbitind=ip_table[i]-1;// IP table is 1-based index

        int srcbit=(int)((input >>(64-srcbitind))&1);//get the bit from input
        output|=(long)srcbit<<(63-i); //set the bit in output;
    }
    System.out.println("Initial permutation: "+Long.toHexString(output));
   }

}
```

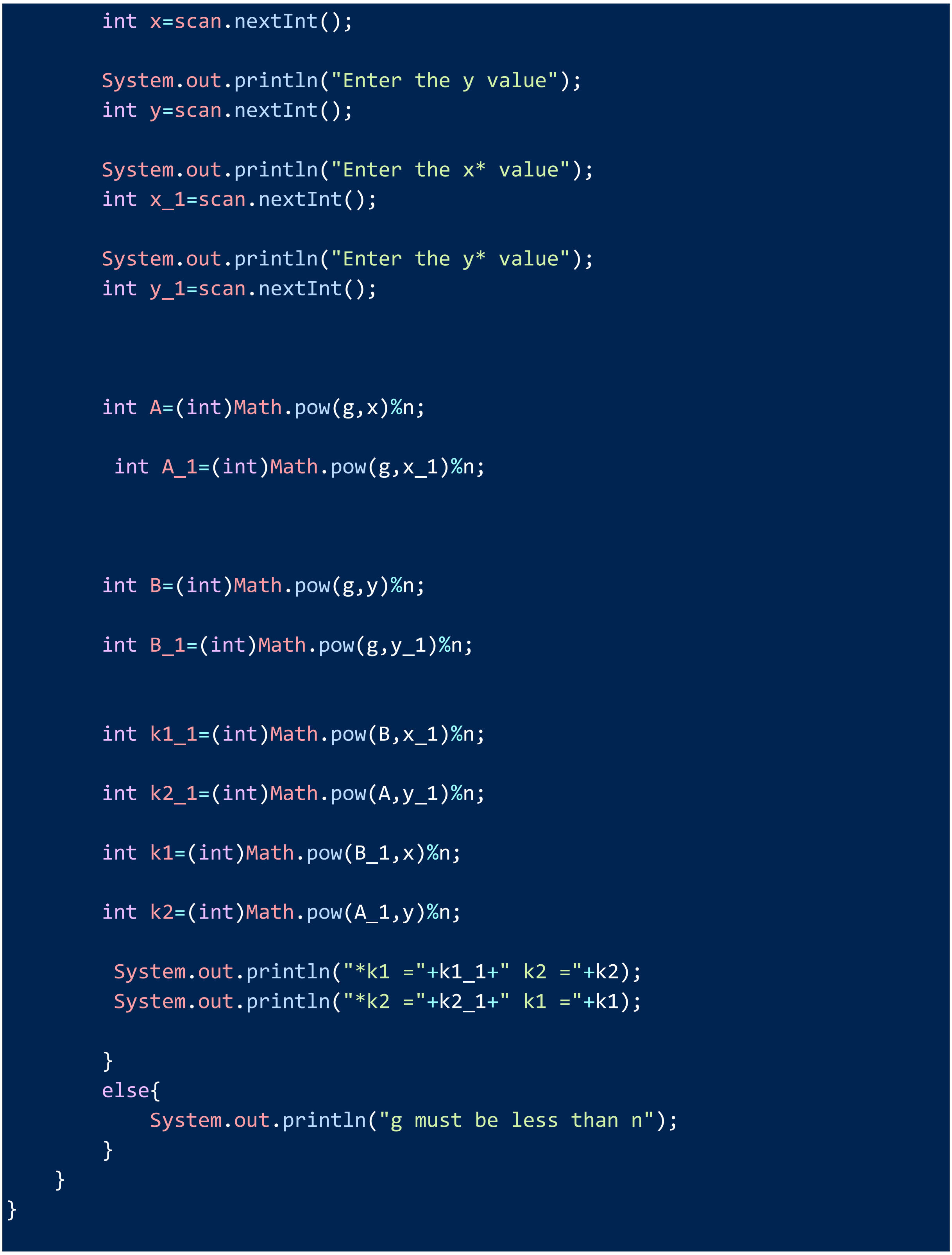
Output:



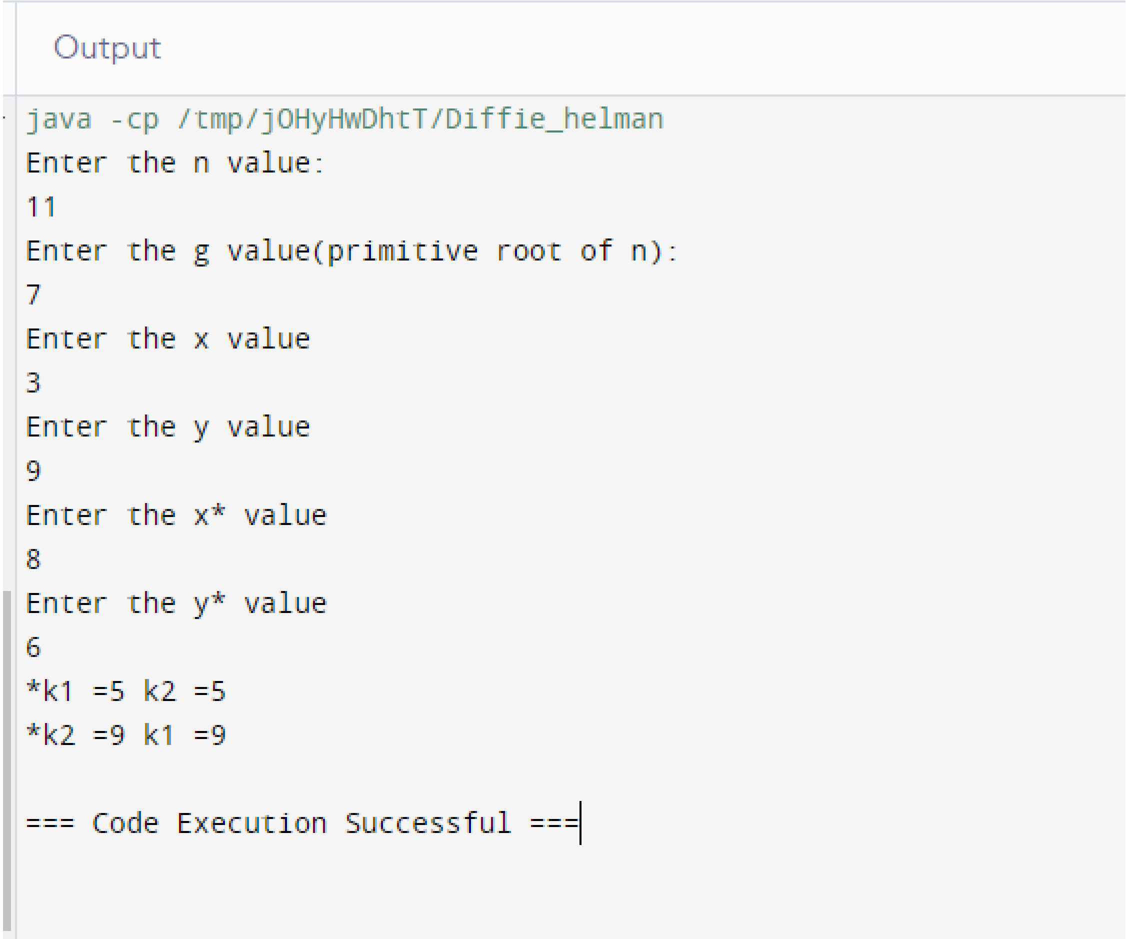
```
java -cp /tmp/QOxBcquMJF/InitialPermutationDES
Enter the input :
123456789abcdef
Initial permutation: f0aaf0aaffcc00cc

=== Code Execution Successful ===
```

16.Diffie-Helman with attack:

```
import java.util.*;
class Diffie_helman {
    public static void main(String[] args) {
        Scanner scan=new Scanner(System.in);
        System.out.println("Enter the n value:");
        int n=scan.nextInt();

        System.out.println("Enter the g value(primitive root of n):");
        int g=scan.nextInt();

        if(g<n){

        System.out.println("Enter the x value");
```



```
        int x=scan.nextInt();

        System.out.println("Enter the y value");
        int y=scan.nextInt();

        System.out.println("Enter the x* value");
        int x_1=scan.nextInt();

        System.out.println("Enter the y* value");
        int y_1=scan.nextInt();



        int A=(int)Math.pow(g,x)%n;

         int A_1=(int)Math.pow(g,x_1)%n;



        int B=(int)Math.pow(g,y)%n;

        int B_1=(int)Math.pow(g,y_1)%n;


        int k1_1=(int)Math.pow(B,x_1)%n;

        int k2_1=(int)Math.pow(A,y_1)%n;

        int k1=(int)Math.pow(B_1,x)%n;

        int k2=(int)Math.pow(A_1,y)%n;

         System.out.println("*k1 ="+k1_1+" k2 ="+k2);
         System.out.println("*k2 ="+k2_1+" k1 ="+k1);

        }
        else{
            System.out.println("g must be less than n");
        }
    }
}
```



Output

```
java -cp /tmp/jOHyHwDhtT/Diffie_helman
Enter the n value:
11
Enter the g value(primitive root of n):
7
Enter the x value
3
Enter the y value
9
Enter the x* value
8
Enter the y* value
6
*k1 =5 k2 =5
*k2 =9 k1 =9

=== Code Execution Successful ===
```