MADRAS INSTITUTE OF TECHNOLOGY

ANNA UNIVERSITY, CHENNAI – 600 044.

B. Tech Information Technology (6/8)

IT7612 –Information Security and Mobile Computing Laboratory

Prepared by: Dr. D. Sangeetha

NDEX

4.	Simplified Data Encryption Standard
5.	Advanced Encryption Standard
6.	RC4
7.	Deffie Hellman key exchange cryptosystem
8.	Rivest-Shamir-Adleman
9.	Elgammal Signature
10.	Secured Hash Algorithm – 512
11.	Digital Signature Standard
12.	Graphical Primitives and Animations in Android.
13.	Event Handling
14.	Location Based Services
15.	Accessing Hardware Devices-Bluetooth
16.	Frequency Reuse

Ex.No:1 SUBSTITUTION CIPHERS

AIM:

To write a program to implement basic substitution ciphers and encrypt the given plaintext and decrypt the same.

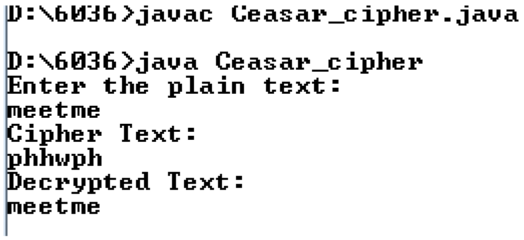
ALGORITHM:

- Start the program.
- Get the plaintext and the required values from the user to encrypt the data.
- Now, encrypt the plaintext given using the appropriate details given by the user any by using the encryption formula for the particular cipher.
- Generate the cipher text.
- Now decrypt the cipher text using the decryption formula for the appropriate cipher and generate the plaintext back.
- Compare the plaintext and the decrypted value.
- Stop.

CEASER CIPHER:

```
import java.util.Scanner;

public class Ceasar_cipher

{

static String alphabet="abcdefghijklmnopqrstuvwxyz";

public static void main(String arg[])

{

Scanner sc=new Scanner(System.in);

String plain,cipher,decipher;

System.out.println("Enter the plain text:");
plain=sc.nextLine();
cipher="";
decipher="";
plain=plain.toLowerCase();
for(int i=0;i<plain.length();i++)
{
cipher+=alphabet.charAt((alphabet.indexOf(plain.charAt(i))+3)%26);
}
System.out.println("Cipher Text:");
System.out.println(cipher);
for(int i=0;i<cipher.length();i++)
{
int check=(alphabet.indexOf(cipher.charAt(i))-3)%26;
if(check<0)
check=26+check;
decipher+=alphabet.charAt(check);
}
System.out.println("Decrypted Text:");
System.out.println(decipher);
}
}
```

OUTPUT:



```
D:\6036>javac Ceasar_cipher.java

D:\6036>java Ceasar_cipher
Enter the plain text:
meetme
Cipher Text:
phhwph
Decrypted Text:
meetme
```

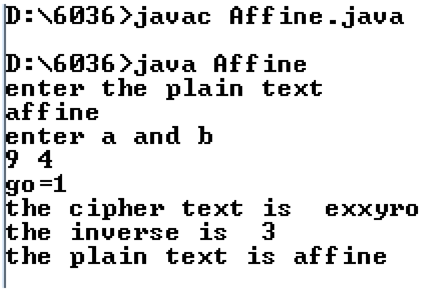
AFFINE CIPHER:

```
import java.io.*;

import java.util.*;

public class Affine

{

public static void main(String args[])

{

int x=0;

Scanner sc=new Scanner(System.in);

System.out.println("enter the plain text");

String input=sc.next();

System.out.println("enter a and b");

int a=sc.nextInt();

int b=sc.nextInt();

int go=gcd(a,26);

System.out.println("go="+go);

if(go==1)

{

String alpha="abcdefghijklmnopqrstuvwxyz";

StringBuilder sb=new StringBuilder();
StringBuilder sb1=new StringBuilder();
for(int i=0;i<input.length();i++)
{
char c=input.charAt(i);
int y=(a*(alpha.indexOf(c))+b)%26;
sb.append(alpha.charAt(y));
}
System.out.println("the cipher text is  "+sb);
int invert=inver(a);
System.out.println("the inverse is  " +invert);
for(int k=0;k<sb.length();k++)
{
char ch=sb.charAt(k);
int mid=invert*(alpha.indexOf(ch)-b);
if(mid<0)
x=26+mid;
else
x=(invert*(alpha.indexOf(ch)-b))%26;
sb1.append(alpha.charAt(x));
}
System.out.println("the plain text is "+sb1);
}
else
```

```
{
System.out.println("the affine cipher is not possible");
}
}
public static int inver(int a)
{
int ans=0;
for(int j=0;j<26;j++)
{
if((a*(j)%26)==1){
ans=j;
break;
}
}
return ans;
}
public static int gcd(int a,int b)
{
if(a==0||b==0)
return 0;
if(a==b)
return a;
if(a>b)
```

```
return gcd(a-b,b);

return gcd(a,b-a);

}}
```

OUTPUT:



```
D:\6036>javac Affine.java

D:\6036>java Affine
enter the plain text
affine
enter a and b
9 4
go=1
the cipher text is  exxyro
the inverse is  3
the plain text is affine
```

PLAYFAIR CIPHER:

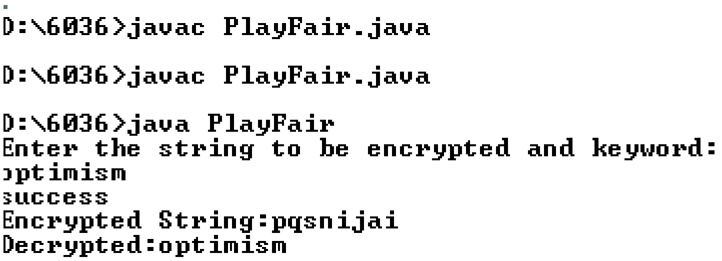
```
package Security;
import java.util.Scanner;
public class PlayFair
{
    static String alphabet="abcdefghijklmnopqrstuvwxyz";
    public static void main(String[] args)
    {
        Scanner sc=new Scanner(System.in);
        String plaintext,keyword;
        System.out.println("Enter the string to be encrypted and keyword:");
        plaintext=sc.nextLine();
        keyword=sc.nextLine();
        plaintext=plaintext.toLowerCase();
        int a[][]=new int[5][5];
        for(int i=0;i<5;i++)
            for(int j=0;j<5;j++)
                a[i][j]=-1;
        String fill="";
        for(int i=0;i<keyword.length();i++)
        {
            if(fill.indexOf(keyword.charAt(i))==-1)
                fill+=keyword.charAt(i);
        }
        for(int i=0;i<alphabet.length()-1;i++)
        {
            if(fill.indexOf(alphabet.charAt(i))==-1)
                fill+=alphabet.charAt(i);
        }
        //System.out.println(fill+" "+fill.length());
        int k=0;
        for(int i=0;i<5;i++)
            for(int j=0;j<5;j++)
            {
                a[i][j]=alphabet.indexOf(fill.charAt(k));
                k++;
                //System.out.print(a[i][j]+" ");
            }
        String diagram="";
        for(int i=0;i<plaintext.length()-1;i=i+2)
        {
            if(plaintext.charAt(i)!=plaintext.charAt(i+1))
            {
                diagram+=plaintext.charAt(i);
                diagram+=plaintext.charAt(i+1);
            }
            else
            {
                diagram+=plaintext.charAt(i);
                diagram+="xx";
                diagram+=plaintext.charAt(i);
            }
        }
        if(plaintext.length()%2==1)
        {
            diagram+=plaintext.charAt(plaintext.length()-1);
            diagram+='x';
        }
```

```
//System.out.println(diagram);
String cipher="";
for(k=0;k<diagram.length();k=k+2)
{
    int b,c;
    b=alphabet.indexOf(diagram.charAt(k));
    c=alphabet.indexOf(diagram.charAt(k+1));
    int x1=0,x2=0,y1=0,y2=0,cn=0;
    for(int i=0;i<5;i++)
        for(int j=0;j<5;j++)
        {
            if(a[i][j]==b)
            {
                x1=i;
                y1=j;
                cn++;
            }
            if(a[i][j]==c)
            {
                x2=i;
                y2=j;
                cn++;
            }
            if(cn==2)
                break;
        }
    if(b==25)
    {
        x1=4;
        y1=4;
    }
    if(c==25)
    {
        x2=4;
        y2=4;
    }
    if(x1==x2)
    {
        cipher+=alphabet.charAt(a[x1][(y1+1)%5]);
        cipher+=alphabet.charAt(a[x2][(y2+1)%5]);
    }
    else if(y1==y2)
    {
        cipher+=alphabet.charAt(a[(x1+1)%5][y1]);
        cipher+=alphabet.charAt(a[(x2+1)%5][y2]);
    }
    else
    {
        cipher+=alphabet.charAt(a[x1][y2]);
        cipher+=alphabet.charAt(a[x2][y1]);
    }
}
System.out.println("Encrypted String:"+cipher);
String decrypt="";
for(k=0;k<cipher.length();k=k+2)
{
    int b,c;
    b=alphabet.indexOf(cipher.charAt(k));
    c=alphabet.indexOf(cipher.charAt(k+1));
    int x1=0,x2=0,y1=0,y2=0,cn=0,X,Y;
    for(int i=0;i<5;i++)
        for(int j=0;j<5;j++)
        {
            if(a[i][j]==b)
            {
                x1=i;
                y1=j;
                cn++;
            }
            if(a[i][j]==c)
            {
                x2=i;
            y2=j;
            cn++;
        }
        if(cn==2)
            break;
    }
if(b==25)
{
    x1=4;
    y1=4;
}
if(c==25)
{
    x2=4;
    y2=4;
}
if(x1==x2)
{
    Y=(y1-1)%5;
    if(Y<0)
        Y=Y+5;
    decrypt+=alphabet.charAt(a[x1][Y]);
    Y=(y2-1)%5;
    if(Y<0)
        Y=Y+5;
    decrypt+=alphabet.charAt(a[x2][Y]);
}
else if(y1==y2)
{
    X=(x1-1)%5;
    if(X<0)
        X=X+5;
    decrypt+=alphabet.charAt(a[X][y1]);
    X=(x2-1)%5;
    if(X<0)
        X=X+5;
```

```
                decrypt+=alphabet.charAt(a[X][y2]);
            }
            else
            {
                decrypt+=alphabet.charAt(a[x1][y2]);
                decrypt+=alphabet.charAt(a[x2][y1]);
            }
        }
        //System.out.println("Decrypted:"+decrypt);
        String decrypt1=decrypt;
        decrypt="";
        for(int i=0;i<decrypt1.length();i++)
        {
            if(i!=decrypt1.length()-
1&&decrypt1.charAt(i)=='x'&&decrypt1.charAt(i+1)=='x')
            {
                i++;
                continue;
            }
            //if(i==(decrypt1.length()-1)&&decrypt1.charAt(i)=='x')
                //break;
            decrypt+=decrypt1.charAt(i);
        }

        //System.out.println("Decrypted:"+decrypt);

        if(decrypt.charAt(decrypt.length()-1)=='x')
        {
            int l=decrypt.length();
            char[] temp = decrypt.toCharArray();
            decrypt="";
            for(int i=0;i<l-1;i++)
                decrypt+=temp[i];
        }
        System.out.println("Decrypted:"+decrypt);
```

```
   }
}
```

OUTPUT:



```
D:\6036>javac PlayFair.java

D:\6036>javac PlayFair.java

D:\6036>java PlayFair
Enter the string to be encrypted and keyword:
optimism
success
Encrypted String:pqsnijai
Decrypted:optimism
```

MULTIPLICATIVE CIPHER:

```
import java.io.*;
import java.util.*;
public class Ciphers
{
public static void main(String args[])
{
int ans1,l,dup=0;
Scanner sc=new Scanner(System.in);
String alpha="abcdefghijklmnopqrstuvwxyz";
do
{
System.out.println("enter choice 1.multiplicative cipher 2. .polyalphabetic cipher
3.monoalphabetic cipher");
int choice=sc.nextInt();
switch(choice)
{
case 1:
{
System.out.println("enter the plain text");
String input=sc.next();
System.out.println("enter the key");
int key=sc.nextInt();
```

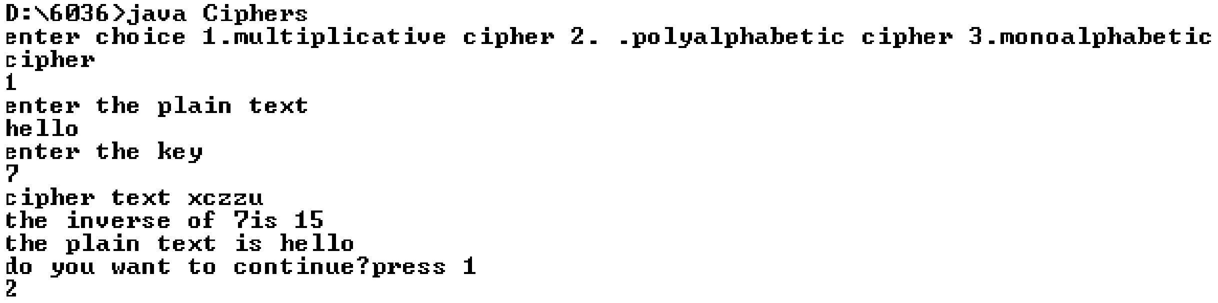
```
StringBuilder sb=new StringBuilder();
StringBuilder sb1=new StringBuilder();
for(int i=0;i<input.length();i++)
{
int y=alpha.indexOf(input.charAt(i));
char ch=alpha.charAt((y*key)%26);
sb.append(ch);
}
System.out.println("cipher text "+sb);
int invert=inver(key);
System.out.println("the inverse of "+key+"is "+invert);
for(int j=0;j<sb.length();j++)
{
int k=(alpha.indexOf(sb.charAt(j))*invert)%26;
sb1.append(alpha.charAt(k));
}
System.out.println("the plain text is "+sb1);
break;
}
case 2:
{
System.out.println("enter the plain text");
String input1=sc.next();
System.out.println("enter the known key");
int key1=sc.nextInt();
int array1[]=new int[input1.length()];
int array2[]=new int[input1.length()];
int array3[]=new int[input1.length()];
for(int i=0;i<input1.length();i++)
{
array1[i]=alpha.indexOf(input1.charAt(i));
//System.out.println(array1[i]);
}
array2[0]=key1;
for(int i=1;i<input1.length();i++)
{
```

```
array2[i]=array1[i-1];
//System.out.println(array2[i]);
}
StringBuilder sb2=new StringBuilder();
StringBuilder sb3=new StringBuilder();
for(int i=0;i<input1.length();i++)
{
int x=(array1[i]+array2[i])%26;
array3[i]=x;
char cha=alpha.charAt(x);
sb2.append(cha);
}
System.out.println("the cipher text is "+sb2);
//System.out.println(input1.length());
//System.out.println(sb2.length());
for(int i=0;i<input1.length();i++)
{
//System.out.println(sb2.charAt(i));
//System.out.println(array2[i]);
//System.out.println(array3[i]);
int x=array3[i]-array2[i];
//System.out.println("x"+x);
if(x<0){
l=26+x;}
else{
l=x;}
//System.out.println("l"+l);
int z=l%26;
//System.out.println(alpha.charAt(l));
char chars=alpha.charAt(z);
sb3.append(chars);
}
System.out.println("the plain text is "+sb3);
break;
}
case 3:
```

```
{
System.out.println("enter the plain text");
String input3=sc.next();
String known="qrrdqr";
char ar[]=known.toCharArray();
int res=0,counter=0;
for(int i=0;i<input3.length();i++)
{
res=0;
char c=known.charAt(i);
for(int j=i+1;j<input3.length();j++)
{
if(known.charAt(j)==c)
{
System.out.println("duplicate occurance of "+c+" enter new value");
char c1=sc.next().charAt(0);
ar[j]=c1;
}
/*for(int k=0;k<ar.length;k++)
{
System.out.println(ar[k]);
}*/
}
}
System.out.println("the cipher text is:");
for(int i=0;i<ar.length;i++)
{
System.out.print(ar[i]);
}
System.out.println(" ");

System.out.println("the plain text is "+input3);
break;
}
default:
throw new IllegalArgumentException("no proper choice");
```

```
}
System.out.println("do you want to continue?press 1");
ans1=sc.nextInt();
}while(ans1==1);
}
public static int inver(int a)
{
int ans=0;
for(int j=0;j<26;j++)
{
if((a*(j)%26)==1){
ans=j;
break;
}
}
return ans;
}
}
```

OUTPUT:



```
D:\6036>java Ciphers
enter choice 1.multiplicative cipher 2. .polyalphabetic cipher 3.monoalphabetic
cipher
1
enter the plain text
hello
enter the key
7
cipher text xczzu
the inverse of 7is 15
the plain text is hello
do you want to continue?press 1
2
```

HILL CIPHER:

```
package Security;
import java.util.Scanner;
public class HillCipher
{
    static String alphabet="abcdefghijklmnopqrstuvwxyz";
    public static int check(int a)
    {
```

```
    while(a<0)
        a+=26;
    return a;
}
public static int[][] multiply(int r,int nc,int a[][],int b[][])
{
    int c[][]=new int[r][nc];
    for(int i=0;i<r;i++)
        for(int j=0;j<nc;j++)
        {
            c[i][j]=0;
            for(int k=0;k<r;k++)
            {
                c[i][j]+=(a[i][k]*b[k][j]);
            }
            c[i][j]%=26;
            while(c[i][j]<0)
                c[i][j]+=26;
        }
    return c;
}
public static int gcd(int a,int b)
{
    if(b==0)
        return a;
    return gcd(b,a%b);
}
public static int inverse(int a)
{
    int q,r1=26,r2=a,r,t1=0,t2=1,t;
    while(r2>0)
    {
        q=r1/r2;
        r=r1%r2;
        t=t1-q*t2;
        t1=t2;
        t2=t;
        r1=r2;
        r2=r;
    }
    if(t1<0)
        t1=t1+26;
    return t1;
}
public static void main(String[] args)
{
    Scanner sc=new Scanner(System.in);
    String plaintext,encrypt,decrypt;
    System.out.println("Enter the plaintext to be encrypted:");
    plaintext=sc.nextLine();
    encrypt="";
    decrypt="";
    int r=0,c=0,kd=0,ch=1;
    while(ch==1)
    {
        System.out.println("Enter the row and column of key matrix:");
        r=sc.nextInt();
        c=sc.nextInt();
        if((r==2&&c==2)||(r==3&&c==3))
                ch=0;
        else
            System.out.println("Enter the valid row and column");
    }
    int k[][],ki[][],kdi;
    if(r==2)
        k=new int[2][2];
    else
        k=new int[3][3];
    ch=1;
    while(ch==1)
    {
        if(r==2&&c==2)
```

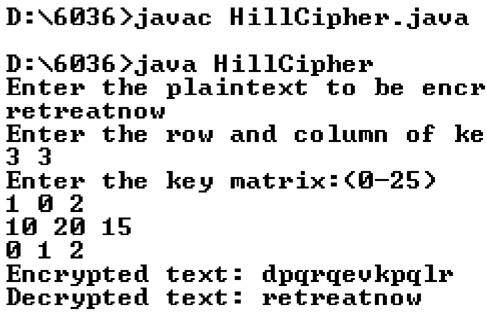
```
            {

                System.out.println("Enter the key matrix:(0-25)");
                for(int i=0;i<r;i++)
                    for(int j=0;j<c;j++)
                        k[i][j]=sc.nextInt();
                kd=(k[0][0]*k[1][1])-(k[0][1]*k[1][0]);
                kd%=26;
                while(kd<0)
                    kd+=26;
                //System.out.println("d: "+kd);

                if(gcd(kd,26)==1)
                    ch=0;
                else
                    System.out.println("Enter a valid key matrix");
            }
            else if(r==3&&c==3)
            {
                System.out.println("Enter the key matrix:(0-25)");
                for(int i=0;i<r;i++)
                    for(int j=0;j<c;j++)
                        k[i][j]=sc.nextInt();
                kd=(k[0][0]*k[1][1]*k[2][2])-
(k[0][0]*k[1][2]*k[2][1])+(k[0][1]*k[1][2]*k[2][0])-
(k[0][1]*k[1][0]*k[2][2])+(k[0][2]*k[1][0]*k[2][1])-(k[0][2]*k[1][1]*k[2][0]);
                kd%=26;
                while(kd<0)
                    kd+=26;
                if(gcd(kd,26)==1)
                    ch=0;
                else
                    System.out.println("Enter a valid key matrix");
                //System.out.println("d:"+kd);
                //System.out.println("d:"+(kd%26));
            }
```

```
}
int p[][],e[][],d[][];
if(r==2&&c==2)
{
    int nc;
    if(plaintext.length()%2==0)
    {
        nc=plaintext.length()/2;
        p=new int[2][nc];
    }
    else
    {
        nc=plaintext.length()/2+1;
        p=new int[2][nc];
    }
    //System.out.println(nc);
    int l=0;
    for(int i=0;i<nc;i++)
        for(int j=0;j<r;j++)
        {
            if(i==nc-1&&j==r-1&&(plaintext.length()%2==1))
                p[j][i]=alphabet.indexOf('x');
            else
                p[j][i]=alphabet.indexOf(plaintext.charAt(l++));
                //System.out.print(p[j][i]+" ");
        }
    //System.out.println("\n");
    e=new int[r][nc];
    e=multiply(r,nc,k,p);
    for(int i=0;i<nc;i++)
        for(int j=0;j<r;j++)
        {
            encrypt+=alphabet.charAt(e[j][i]);
        }
    System.out.println("Encrypted text: "+encrypt);
    ki=new int[2][2];
    kdi=inverse(kd);
    //System.out.println("di: "+kdi);
    ki[0][0]=kdi*k[1][1]%26;
    ki[1][1]=kdi*k[0][0]%26;
    ki[0][1]=kdi*(-k[0][1])%26;
    ki[1][0]=kdi*(-k[1][0])%26;
    d=new int[r][nc];
    d=multiply(r,nc,ki,e);
    for(int i=0;i<nc;i++)
        for(int j=0;j<r;j++)
        {
            decrypt+=alphabet.charAt(d[j][i]);
        }
    if(decrypt.charAt(decrypt.length()-1)=='x')
    {
        int len=decrypt.length();
        char[] temp = decrypt.toCharArray();
        decrypt="";
        for(int i=0;i<len-1;i++)
            decrypt+=temp[i];
    }
    System.out.println("Decrypted text: "+decrypt);
}
else if(r==3&&c==3)
{
    int nc;
    if(plaintext.length()%3==0)
    {
        nc=plaintext.length()/3;
        p=new int[3][nc];
    }
    else
    {
        nc=plaintext.length()/3+1;
        p=new int[3][nc];
    }
```

```
            //System.out.println(nc);
            int l=0;
            for(int i=0;i<nc;i++)
                for(int j=0;j<r;j++)
                {
                    //System.out.println("i:"+i+" j:"+j);
                    if(i==nc-1&&j==r-
1&&((plaintext.length()%3==2)||(plaintext.length()%3==1)))
                        p[j][i]=alphabet.indexOf('x');
                    else if(i==nc-1&&j==r-2&&(plaintext.length()%3==1))
                        p[j][i]=alphabet.indexOf('x');
                    else
                        p[j][i]=alphabet.indexOf(plaintext.charAt(l++));
                        //System.out.print(p[j][i]+" ");
                }
            //System.out.println("\n");
            e=new int[r][nc];
            e=multiply(r,nc,k,p);
            for(int i=0;i<nc;i++)
                for(int j=0;j<r;j++)
                {
                    encrypt+=alphabet.charAt(e[j][i]);
                }
            System.out.println("Encrypted text: "+encrypt);
            ki=new int[3][3];
            kdi=inverse(kd);
            //System.out.println("di: "+kdi);
            ki[0][0]=kdi*((k[1][1]*k[2][2])-(k[1][2]*k[2][1]))%26;
            ki[0][1]=kdi*((k[0][2]*k[2][1])-(k[0][1]*k[2][2]))%26;
            ki[0][2]=kdi*((k[0][1]*k[1][2])-(k[0][2]*k[1][1]))%26;
            ki[1][0]=kdi*((k[1][2]*k[2][0])-(k[1][0]*k[2][2]))%26;
            ki[1][1]=kdi*((k[0][0]*k[2][2])-(k[0][2]*k[2][0]))%26;
            ki[1][2]=kdi*((k[0][2]*k[1][0])-(k[0][0]*k[1][2]))%26;
            ki[2][0]=kdi*((k[1][0]*k[2][1])-(k[1][1]*k[2][0]))%26;
            ki[2][1]=kdi*((k[0][1]*k[2][0])-(k[0][0]*k[2][1]))%26;
            ki[2][2]=kdi*((k[0][0]*k[1][1])-(k[0][1]*k[1][0]))%26;
```

```
    for(int i=0;i<3;i++)
        for(int j=0;j<3;j++)
        {
            if(ki[i][j]<0)
                ki[i][j]=check(ki[i][j]);
            //System.out.print(ki[i][j]+" ");
        }
    d=new int[r][nc];
    d=multiply(r,nc,ki,e);
    for(int i=0;i<nc;i++)
        for(int j=0;j<r;j++)
        {
            decrypt+=alphabet.charAt(d[j][i]);
        }
    if(decrypt.charAt(decrypt.length()-1)=='x')
    {
        int len=decrypt.length();
        char[] temp = decrypt.toCharArray();
        if(decrypt.charAt(decrypt.length()-2)=='x')
        {
            decrypt="";
            for(int i=0;i<len-2;i++)
                decrypt+=temp[i];
        }
        else
        {
            decrypt="";
            for(int i=0;i<len-1;i++)
                decrypt+=temp[i];
        }
    }
    System.out.println("Decrypted text: "+decrypt);
 }}}
```

OUTPUT:

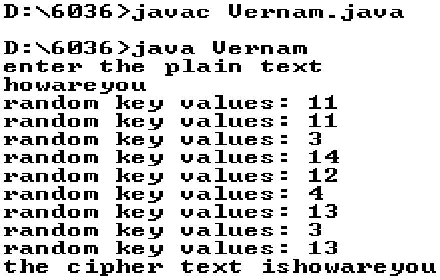


```
D:\6036>javac HillCipher.java

D:\6036>java HillCipher
Enter the plaintext to be encrypted:
retreatnow
Enter the row and column of key matrix:
3 3
Enter the key matrix:(0-25)
1 0 2
10 20 15
0 1 2
Encrypted text: dpqrqevkpqlr
Decrypted text: retreatnow
```

VERNAM CIPHER:

```
import java.io.*;
import java.util.*;
public class Vernam
{
public static void main(String args[])
{
int x=0;
Scanner sc=new Scanner(System.in);
Random random=new Random();
System.out.println("enter the plain text");
String input=sc.next();
StringBuilder sb=new StringBuilder();
String alpha="abcdefghijklmnopqrstuvwxyz";
int l=input.length();
int arr[]=new int[l];
int arr1[]=new int[l];
for(int i =0; i<l;i++){
    arr[i] = random.nextInt(15);
    System.out.println("random key values: " + arr[i]);
}
for(int i=0;i<l;i++){
arr1[i]=alpha.indexOf(input.charAt(i))+arr[i];
if(arr1[i]>26){
x=arr1[i]-26;}
else{
x=arr1[i];}
sb.append(alpha.charAt(x));}
System.out.println("the cipher text is"+sb);
}}
```

OUTPUT:

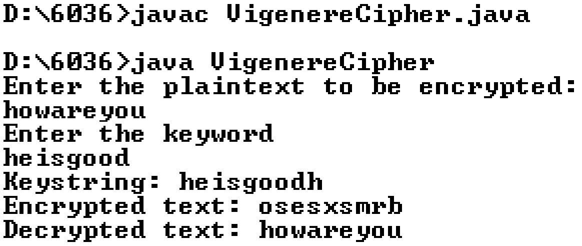


```
D:\6036>javac Vernam.java

D:\6036>java Vernam
enter the plain text
howareyou
random key values: 11
random key values: 11
random key values: 3
random key values: 14
random key values: 12
random key values: 4
random key values: 13
random key values: 3
random key values: 13
the cipher text ishowareyou
```

VIGNERE TABLEU CIPHER:

```
import java.util.Scanner;
public class VigenereCipher
{
    static String alphabet="abcdefghijklmnopqrstuvwxyz";
    public int[][] k;
    public VigenereCipher()
    {
        k=new int[26][26];
        int l=0;
        for(int i=0;i<26;i++)
        {
            l=i;
            for(int j=0;j<26;j++)
            {


                k[i][j]=l;
                //System.out.print(k[i][j]+" ");
                l=((++l)%26);
            }
            //System.out.println();
        }
    }
    public static void main(String[] args)
    {
        VigenereCipher vc=new VigenereCipher();
        Scanner sc=new Scanner(System.in);
        String plaintext,encrypt,decrypt,keyword;
        System.out.println("Enter the plaintext to be encrypted:");
        plaintext=sc.nextLine();
        System.out.println("Enter the keyword");
        keyword=sc.nextLine();
        String keystring="";
        int j=0;
        for(int i=0;i<plaintext.length();i++)
        {
            keystring+=keyword.charAt(j);
            if(j<keyword.length()-1)
                j++;
            else
                j=0;
        }
        System.out.println("Keystring: "+keystring);
        encrypt="";
        for(int i=0;i<plaintext.length();i++)
        {

encrypt+=alphabet.charAt(vc.k[alphabet.indexOf(plaintext.charAt(i))][alpha
bet.indexOf(keystring.charAt(i))]);
        }
        System.out.println("Encrypted text: "+encrypt);
        decrypt="";
        for(int i=0;i<keystring.length();i++)
        {
            int l=alphabet.indexOf(keystring.charAt(i));
            int s=alphabet.indexOf(encrypt.charAt(i));
            for(int k=0;k<26;k++)
            {
                if(vc.k[l][k]==s)
                {
                    decrypt+=alphabet.charAt(k);
                    break;
                }
            }
        }
        System.out.println("Decrypted text: "+decrypt);
    }
}
```

OUTPUT:



```
D:\6036>javac VigenereCipher.java

D:\6036>java VigenereCipher
Enter the plaintext to be encrypted:
howareyou
Enter the keyword
heisgood
Keystring: heisgoodh
Encrypted text: osesxsmrb
Decrypted text: howareyou
```

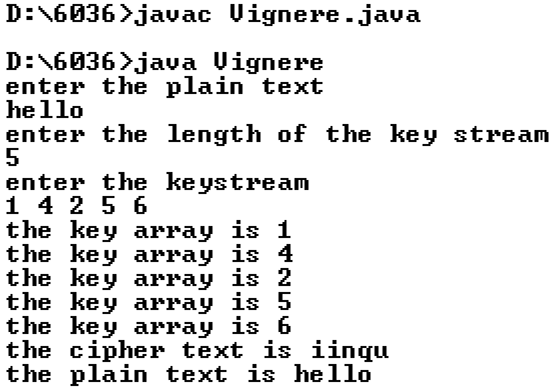
VIGNERE CIPHER:

```
import java.io.*;
import java.util.*;
public class Vignere
{
public static void main(String args[])
{
StringBuilder sb=new StringBuilder();
StringBuilder sb1=new StringBuilder();
Scanner sc=new Scanner(System.in);
System.out.println("enter the plain text");
```

```
String input=sc.next();
System.out.println("enter the length of the key stream");
int keyl=sc.nextInt();
int key[]=new int[keyl];
System.out.println("enter the keystream");
for(int i=0;i<keyl;i++)
{
key[i]=sc.nextInt();
}
String alpha="abcdefghijklmnopqrstuvwxyz";
int l=input.length();
int arr[]=new int[l];
int arr1[]=new int[l];
int arr2[]=new int[l];
for(int i=0;i<l;i++)
{
arr[i]=alpha.indexOf(input.charAt(i));
}
int k=0;
for(int i=0;i<l;i++)
{
arr1[i]=key[k];
System.out.println("the key array is "+arr1[i]);
if(k<keyl-1){
k++;}
else{
k=0;}
}
for(int i=0;i<l;i++)
{
arr2[i]=arr1[i]+arr[i];
sb.append(alpha.charAt(arr2[i]%26));
}
```

```
System.out.println("the cipher text is "+sb);
for(int i=0;i<l;i++)
{
int x=(arr2[i]-arr1[i]);
sb1.append(alpha.charAt(x));
}
System.out.println("the plain text is "+sb1);
}
}
```

OUTPUT:



```
D:\6036>javac Vignere.java

D:\6036>java Vignere
enter the plain text
hello
enter the length of the key stream
5
enter the keystream
1 4 2 5 6
the key array is 1
the key array is 4
the key array is 2
the key array is 5
the key array is 6
the cipher text is iinqu
the plain text is hello
```

RESULT:

Thus basic substitution ciphers have been implemented successfully.

EX.NO: 2

TRANSPOSITION CIPHERS

AIM:

To implement the following transposition ciphers for encryption and decryption of the given message.

(i) Railfence cipher

(ii) (ii) Columnar cipher

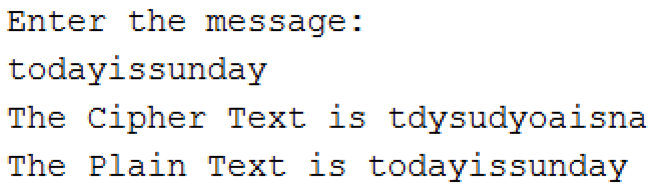
ALGORITHM:

1. Start the program.
2. Obtain the input message from the user.
3. Encrypt the given message using the appropriate transposition mechanisms for the
   respective ciphers.
4. Generate the cipher text.
5. Now decrypt the cipher text using deciphering mechanisms for the respective ciphers.
6. Generate the plain text.
7. Compare the plain text generated and user’s input message.
8. Stop.

RAILFENCE CIPHER:

```
import java.util.*;
public class RailFence {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String PT=sc.next();
        int n=PT.length();
        String CT="";
        for(int i=0;i<n;i+=2){
            CT=CT+PT.charAt(i);
        }
        for(int i=1;i<n;i+=2){
            CT=CT+PT.charAt(i);
        }
        System.out.println("The Cipher Text is "+CT);
        int p = n%2;
            String Pt="";
        if(p==0){
            int j=0;
         for(int i=n/2;i<n;i++){
            Pt=Pt+CT.charAt(j)+CT.charAt(i);
            j++;
        }
        }
        else{
            int j=0;
         for(int i=((n/2)+1);i<n;i++){
            Pt=Pt+CT.charAt(j)+CT.charAt(i);
            j++;
        }
          Pt=Pt+CT.charAt(j);
        }

        System.out.println("The Plain Text is "+Pt);
        }
    }
```

OUTPUT:



```
Enter the message:
todayissunday
The Cipher Text is tdysudyoaisna
The Plain Text is todayissunday
```

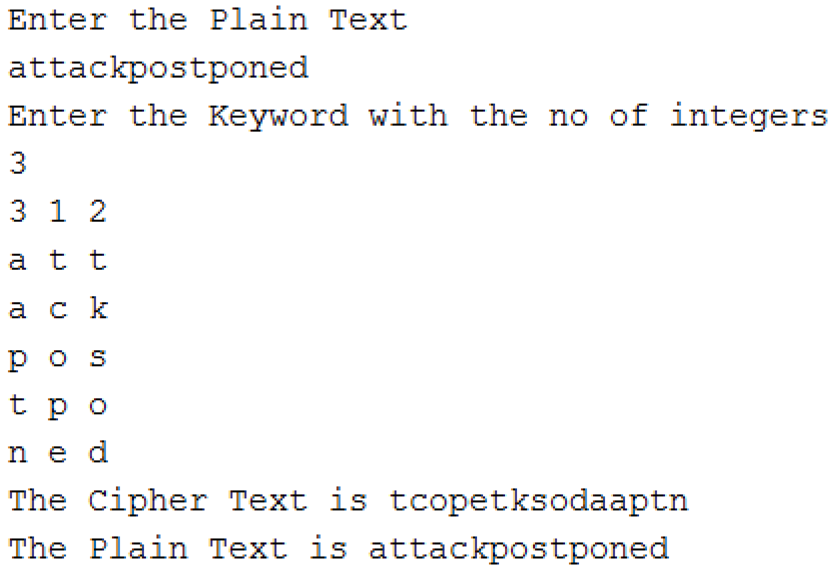
COLUMNAR CIPHER:

```
import java.util.*;
public class Columnar {
    public static void main(String[] args) {
        Scanner scan=new Scanner(System.in);
        System.out.println("Enter the Plain Text ");
        String PT=scan.next();
        PT=PT.replaceAll(" ", "");
```

```
System.out.println("Enter the Keyword with the no of integers ");
int n = scan.nextInt();
int k[]=new int[n];
for(int i=0;i<n;i++)
{
    k[i]=scan.nextInt();
}
int l=PT.length();
if(l%3 != 0){
    l=(l/n)+1;
}else{
     l=l/n;
}

int mat[][]=new int[l][n];
int p=0;
for(int i=0;i<l;i++)
{
    for(int j=0;j<n;j++){
        if(p != PT.length() ){
        mat[i][j]=(PT.charAt(p)-97);
        }else{
            mat[i][j]=-2;
        }
        p++;
    }
}
for (int a=0;a<l;a++){
    for (int j=0;j<n;j++){
        System.out.print((char)(mat[a][j]+97)+" ");
    }
    System.out.println();
}
String CT="";
int s=1,i;
while(s<=n){
    for( i=0;i<n;i++){
  if(k[i]==s){
      for(int j=0;j<l;j++){
       CT+=(char)(mat[j][i]+97);
   }
```

```
            }
          }
           s++;
          }
          System.out.println("The Cipher Text is "+CT);

         PT="";

         for (int a=0;a<l;a++){
              for (int j=0;j<n;j++){
                  char b=(char)(mat[a][j]+97);
                  if(b!='_')
                  {
                      PT+=b;
                  }
              }
          }
         System.out.println("The Plain Text is "+ PT);
      }

  }
```

OUTPUT:



```
Enter the Plain Text
attackpostponed
Enter the Keyword with the no of integers
3
3 1 2
a t t
a c k
p o s
t p o
n e d
The Cipher Text is tcopetksodaaptn
The Plain Text is attackpostponed
```

RESULT:

Thus the above mentioned transposition ciphers are implemented successfully for encryption and decryption of the given message.

EX.NO: 3

MODULAR ARITHMETIC

AIM:

To implement the following theorems to find multiplicative inverse or to solve a congruent equation or a set of congruent equations.

(i) Euclid's theorem
(ii) Extended Euclid's theorem
(iii) Fermat's theorem
(iv) Euler's theorem
(v) Chinese's Remainder theorem

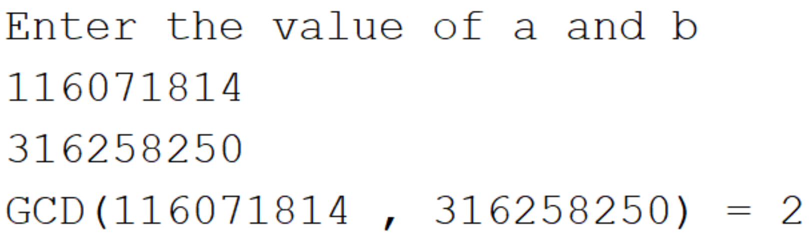
ALGORITHM:

1. Start the program.
2. Obtain the required input values from the user.
3. Perform appropriate operations as per the theorems either to find greatest common divisor or multiplicative inverse or to solve a set of congruency.
4. Print the result.
5. Stop.

EUCLID'S THEOREM:

```
import java.util.*;
public class Euclids {
    public static int GCD(int a,int b)
    {
    if(b == 0)
      return a;
    else
      return GCD(b, a%b);
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int a, b, g;
        System.out.println("Enter the value of a and b ");
        a = sc.nextInt();
        b = sc.nextInt();
        g = GCD(a, b);
        System.out.println("GCD(" + a + " , " + b + ") = " + g);
    }
}
```

OUTPUT:

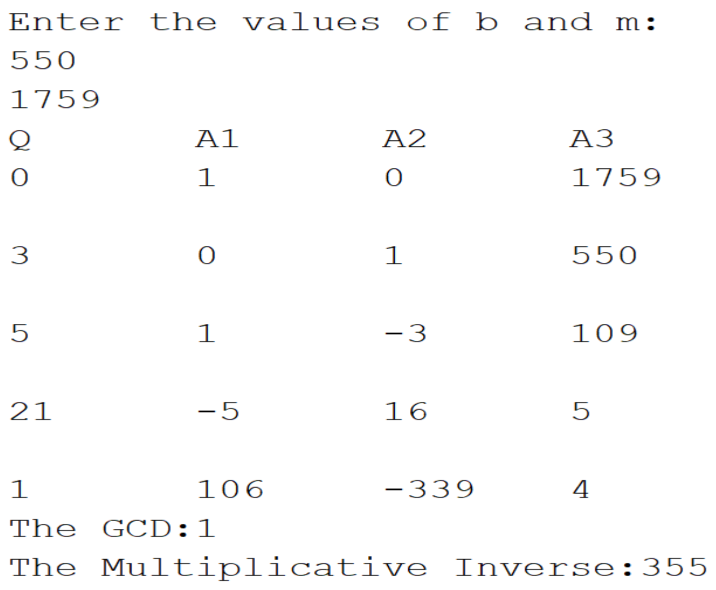


```
Enter the value of a and b
116071814
316258250
GCD(116071814 , 316258250) = 2
```

EXTENDED EUCLID'S THEOREM:

```
import java.util.*;
public class Extendedeuclids {
    public static void main(String[] args) {
    int b,m,gcd,mi,Q=0;
    Scanner s = new Scanner(System.in);
    System.out.println("Enter the values of b and m:");
    b = s.nextInt();
    m = s.nextInt();
    int A1=1,A2=0,B1=0,B2=1,A3=m,B3=b,T1=0,T2=0,T3=0;
System.out.println("Q\tA1\tA2\tA3\tB1\tB2\tB3\n"+Q+"\t"+A1+"\t"+A2+"\t"+A
3+"\t"+B1+"\t"+B2+"\t"+B3);
    cond(A1,A2,A3,B1,B2,B3,T1,T2,T3);
    }
public static void cond(int A1,int A2,int A3,int B1,int B2,int B3,int T1,int T2,int
T3)
    {
    int Q = A3/B3,gcd,mi;
    if(B3==0)
    {
        gcd=A3;
        System.out.println("The GCD:"+gcd);
    }
    else if(B3==1)
    {
        gcd=B3;
        mi=B2;
        System.out.println("The GCD:"+gcd);
        System.out.println("The Multiplicative Inverse:"+mi);
    }
    else
        update(A1,A2,A3,B1,B2,B3,T1,T2,T3,Q);
    }
public static void update(int A1,int A2,int A3,int B1,int B2,int B3,int T1,int
T2,int T3,int Q)
    {
    T1=A1-Q*B1;
    T2=A2-Q*B2;
    T3=A3-Q*B3;
    A1=B1;
    A2=B2;
    A3=B3;
    B1=T1;
    B2=T2;
    B3=T3;

System.out.println("\n"+Q+"\t"+A1+"\t"+A2+"\t"+A3+"\t"+B1+"\t"+B2+"\t"+B3
);
    cond(A1,A2,A3,B1,B2,B3,T1,T2,T3);
    }
}
```

OUTPUT:



```
Enter the values of b and m:
550
1759
Q          A1         A2         A3         B1         B2         B3
0          1          0          1759       0          1          550

3          0          1          550        1          -3         109

5          1          -3         109        -5         16         5

21         -5         16         5          106        -339       4

1          106        -339       4          -111       355        1
The GCD:1
The Multiplicative Inverse:355
```

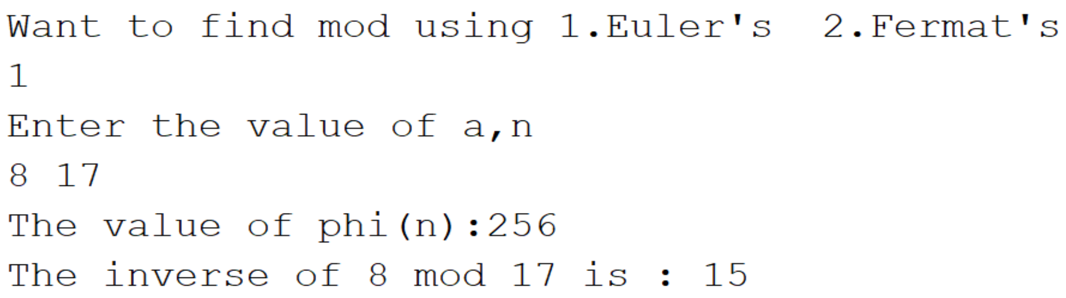
FERMAT'S AND EULER'S THEOREM:

```
import java.util.*;
public class Eulfer {
static Scanner s = new Scanner(System.in);
   public static void main(String[] args) {
      int choice;
      System.out.println("Want to find mod using 1.Euler's  2.Fermat's");
      choice = s.nextInt();
      switch(choice){
         case 1:euler();break;
         case 2:fermat();break;
      }
   }
      public static void euler(){
      int a,n,p=1,i=2,gcd=0,c=0;
      System.out.println("Enter the value of a,n");
      a=s.nextInt();
      n=s.nextInt();
      int k=n;
      gcd = GCD(a,n);
      if(gcd==1){
         if(n==1)
            p=1;
```

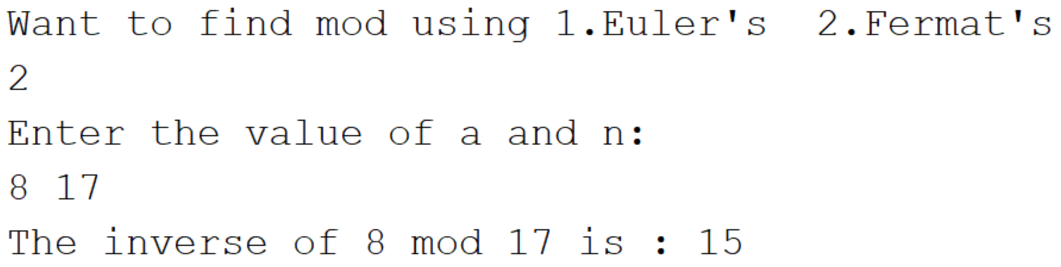
```
        for(int j=2; j<n/2; j++)
            {
        if(n%j == 0)
        {
            c++;
        }
    }
    if(c == 0)
    {
        p=n-1;
    }
    while(n>1){
    if(n%i==0)
    {
        n=n/i;
        p=p*(i-1);
    }
    else
    {
        i++;
    }
}
System.out.println("The value of phi(n):"+p);
p=p-1;
mod(a,p,k);
    }
else
    {
        System.out.println("Condition not satisfied");
    }
        }
public static int GCD(int a,int b)
{
if(b == 0)
  return a;
else
  return GCD(b, a%b);
}
public static void mod(int a,int p,int n)
{
    int x=1;
```

```
        int j=a;
         while (p > 0) {
            if (p % 2 == 1) {
                x = (x * a);
                if (x > n) {
                    x %= n;
                }
            }
            a = (a * a);
            if (a > n) {
                a %= n;
            }
            p /= 2;
        }

        System.out.println("The inverse of "+j+" mod "+n+" is : " + x);
    }
    public static void fermat()
    {
        int a,p,n,i,flag=0;
        System.out.println("Enter the value of a and n:");
        a=s.nextInt();
        n=s.nextInt();
        for(i=2; i<n/2; i++)
        {
            if(n%i == 0)
            {
                System.out.println("Fermat's can't be applied since n is not prime");
                flag=1;
                break;
            }
        }
        if(flag==0){
        p=n-2;
        int j=a;
        int x=1;
         while (p > 0) {
            if (p % 2 == 1) {
                x = (x * a);
                if (x > n) {
                    x %= n;
            }
         }
         a = (a * a);
         if (a > n) {
            a %= n;
         }
         p /= 2;
      }
 System.out.println("The inverse of "+j+" mod "+n+" is : " + x);
   }
   }}
```

OUTPUT:



```
Want to find mod using 1.Euler's  2.Fermat's
1
Enter the value of a,n
8 17
The value of phi(n):256
The inverse of 8 mod 17 is : 15
```



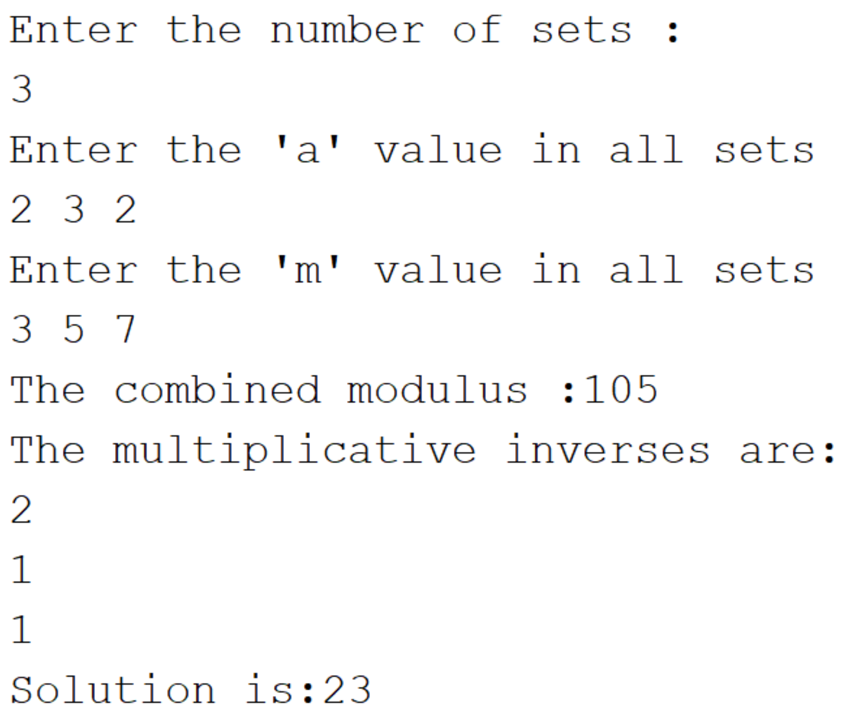
```
Want to find mod using 1.Euler's  2.Fermat's
2
Enter the value of a and n:
8 17
The inverse of 8 mod 17 is : 15
```

CHINESE'S REMAINDER THEOREM:

```
import java.util.*;
public class Chinese {
static Scanner s = new Scanner(System.in);
   public static void main(String[] args) {
      int n,i,M1=1,x=0;
      int a[]=new int[6];
      int m[]=new int[6];
      int M[]=new int[6];
      int mi[]=new int[6];
      System.out.println("Enter the number of sets : ");
```

```
        n=s.nextInt();
        System.out.println("Enter the 'a' value in all sets");
        for(i=0;i<n;i++)
        {
            a[i]=s.nextInt();
        }
        System.out.println("Enter the 'm' value in all sets");
        for(i=0;i<n;i++)
        {
            m[i]=s.nextInt();
        }
        for(i=0;i<n;i++)
        {
            int a1=a[i],g;
            i++;
            int b1=a[i];
            g=GCD(a1,b1);
            if(g==1)
                continue;
        }
        for(i=0;i<n;i++)
        {
            M1=M1*m[i];
        }
        System.out.println("The combined modulus :"+M1);
        System.out.println("The multiplicative inverses are:");
        for(i=0;i<n;i++)
        {
            M[i]=M1/m[i];
            int a1=M[i];
            int n1=m[i];
            int p1=n1-2;
            mi[i]=mod(a1,p1,n1);
            System.out.println(mi[i]);
            x=x+(a[i]*M[i]*mi[i]);
        }
        x= x%M1;
        System.out.println("Solution is:"+x);
    }
    public static int mod(int a,int p,int n)
    {
        int x=1;
        int j=a;
         while (p > 0) {
            if (p % 2 == 1) {
                x = (x * a);
                if (x > n) {
                    x %= n;
                }
            }
            a = (a * a);
            if (a > n) {
                a %= n;
            }
            p /= 2;
        }
        return x;
    }
    public static int GCD(int a,int b)
    {
    int gcd=0;
    if(b == 0)
      return a;
    else
      return GCD(b, a%b);
  }
}
```

OUTPUT:



```
Enter the number of sets :
3
Enter the 'a' value in all sets
2 3 2
Enter the 'm' value in all sets
3 5 7
The combined modulus :105
The multiplicative inverses are:
2
1
1
Solution is:23
```

RESULT:

Thus all the above mentioned theorems are implemented to find either GCD or multiplicative inverse or to solve set of congruencies.

Ex.No:4

SIMPLIFIED DATA ENCRYPTION STANDARD

AIM:

To write a program to implement simplified data encryption standard algorithm.

ALGORITHM:

- Start the program
- Get your plaintext and key
- First apply p10 permute with given algorithm and split the generated p10 with 5 and 5.
- shift once by splitting them by half and then permute p8 in which use shifted once. Generated key is k1.
- Similarly generate the key is k2.
- Now perform encrypt permute plaintext with initial permutation[IP] and then take the right half of IP.
- Perform EX-OR operations and swap the values again do the following steps we get encrypted message.
- Similarly follow the above steps we get the plaintext message.

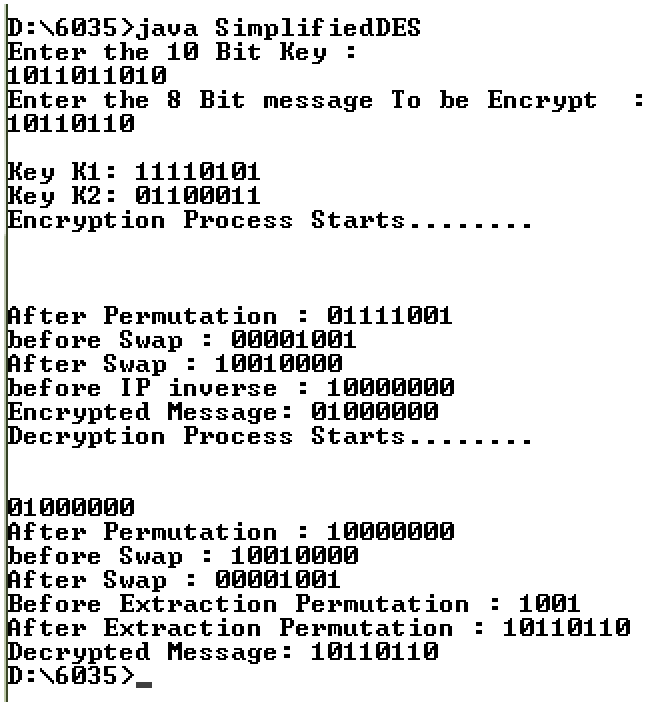
SOURCE CODE:

```
import java.io.*;
import java.lang.*;

class SDES
      {
       public int K1, K2;
       public static final int P10[] = { 3, 5, 2, 7, 4, 10, 1, 9, 8, 6};
       public static final int P10max = 10;
       public static final int P8[] = { 6, 3, 7, 4, 8, 5, 10, 9};
       public static final int P8max = 10;
       public static final int P4[] = { 2, 4, 3, 1};
       public static final int P4max = 4;
       public static final int IP[] = { 2, 6, 3, 1, 4, 8, 5, 7};
       public static final int IPmax = 8;
       public static final int IPI[] = { 4, 1, 3, 5, 7, 2, 8, 6};
       public static final int IPImax = 8;
       public static final int EP[] = { 4, 1, 2, 3, 2, 3, 4, 1};
       public static final int EPmax = 4;
       public static final int S0[][] = {{ 1, 0, 3, 2},{ 3, 2, 1, 0},{ 0, 2, 1,
                                                     3},{ 3, 1, 3, 2}};
       public static final int S1[][] = {{ 0, 1, 2, 3},{ 2, 0, 1, 3},{ 3, 0, 1,
                                                     2},{ 2, 1, 0, 3}};

      public static int permute( int x, int p[], int pmax)
      {
        int y = 0;
        for( int i = 0; i < p.length; ++i)
         {
           y <<= 1;
           y |= (x >> (pmax - p[i])) & 1;
       }
        return y;
      }

      public static int F( int R, int K)
      {
         int t = permute( R, EP, EPmax) ^ K;
         int t0 = (t >> 4) & 0xF;
         int t1 = t & 0xF;
         t0 = S0[ ((t0 & 0x8) >> 2) | (t0 & 1) ][ (t0 >> 1) & 0x3 ];
         t1 = S1[ ((t1 & 0x8) >> 2) | (t1 & 1) ][ (t1 >> 1) & 0x3 ];
          t = permute( (t0 << 2) | t1, P4, P4max);
        return t;

 }

 public static int fK( int m, int K)
       {
           int L = (m >> 4) & 0xF;
           int R = m & 0xF;
           return ((L ^ F(R,K)) << 4) | R;
       }

 public static int SW( int x)
 {
  return ((x & 0xF) << 4) | ((x >> 4) & 0xF);
 }
  public byte encrypt( int m)
      {
        System.out.println("\nEncryption Process Starts........\n\n");
         m = permute( m, IP, IPmax);
        System.out.print("\nAfter Permutation : ");
         printData( m, 8);
         m = fK( m, K1);
         System.out.print("\nbefore Swap : ");
         printData( m, 8);
         m = SW( m);
         System.out.print("\nAfter Swap : ");
         printData( m, 8);
         m = fK( m, K2);
         System.out.print("\nbefore IP inverse : ");
         printData( m, 8);
         m = permute( m, IPI, IPImax);
         return (byte) m;

       }
       public byte decrypt( int m)

       {
         System.out.println("\nDecryption Process Starts........\n\n");
         printData( m, 8);
         m = permute( m, IP, IPmax);
         System.out.print("\nAfter Permutation : ");
         printData( m, 8);
         m = fK( m, K2);
         System.out.print("\nbefore Swap : ");
```

```
    printData( m, 8);
    m = SW( m);
    System.out.print("\nAfter Swap : ");
    printData( m, 8);
    m = fK( m, K1);
    System.out.print("\nBefore Extraction Permutation : ");
    printData( m, 4);
    m = permute( m, IPI, IPImax);
    System.out.print("\nAfter Extraction Permutation : ");
    printData( m, 8);
    return (byte) m;

  }
  public static void printData( int x, int n)
   {
     int mask = 1 << (n-1);
     while( mask > 0)
     {
     System.out.print( ((x & mask) == 0) ? '0' : '1');
     mask >>= 1;
     }
  }
 public SDES( int K)
{
    K = permute( K, P10, P10max);
    int t1 = (K >> 5) & 0x1F;
    int t2 = K & 0x1F;
    t1 = ((t1 & 0xF) << 1) | ((t1 & 0x10) >> 4);
    t2 = ((t2 & 0xF) << 1) | ((t2 & 0x10) >> 4);
    K1 = permute( (t1 << 5)| t2, P8, P8max);
    t1 = ((t1 & 0x7) << 2) | ((t1 & 0x18) >> 3);
    t2 = ((t2 & 0x7) << 2) | ((t2 & 0x18) >> 3);
    K2 = permute( (t1 << 5)| t2, P8, P8max);

  }
```

```
      }

// Main operations
      public class SimplifiedDES
      {

        public static void main( String args[]) throws Exception
        {
         DataInputStream inp=new DataInputStream(System.in);
         System.out.println("Enter the 10 Bit Key :");
          int K = Integer.parseInt(inp.readLine(),2);
          SDES A = new SDES( K);
          System.out.println("Enter the 8 Bit message To be Encrypt  : ");
          int m = Integer.parseInt(inp.readLine(),2);
          System.out.print("\nKey K1: ");
          SDES.printData( A.K1, 8);
          System.out.print("\nKey K2: ");
          SDES.printData( A.K2, 8);
          m = A.encrypt( m);
          System.out.print("\nEncrypted Message: ");
          SDES.printData( m, 8);
          m = A.decrypt( m);
          System.out.print("\nDecrypted Message: ");
          SDES.printData( m, 8);

        }

    }
```

OUTPUT:



```
D:\6035>java SimplifiedDES
Enter the 10 Bit Key :
1011011010
Enter the 8 Bit message To be Encrypt  :
10110110

Key K1: 11110101
Key K2: 01100011
Encryption Process Starts........



After Permutation : 01111001
before Swap : 00001001
After Swap : 10010000
before IP inverse : 10000000
Encrypted Message: 01000000
Decryption Process Starts........


01000000
After Permutation : 10000000
before Swap : 10010000
After Swap : 00001001
Before Extraction Permutation : 1001
After Extraction Permutation : 10110110
Decrypted Message: 10110110
D:\6035>_
```

RESULT:

Thus the simplified data encryption standard algorithm has been implemented successfully.

Ex.No:5

ADVANCED DATA ENCRYPTION STANDARD

AIM:

To write a program to implement advanced encryption standard algorithm.

ALGORITHM:

- Start the program
- Get the input values plaintext and key.
- Convert each character to its hex value and ex-or the key and plaintext and obtain the first state matrix.
- Now substitute the values in the state matrix from the s-box.
- Now do shift rows for this shift the 1st row once,2nd row twice,3rd row thrice(circular rotate).
- For mix columns multiply the value of the state matrix with a fixed matrix .
- Stop the program.

SOURCE CODE:

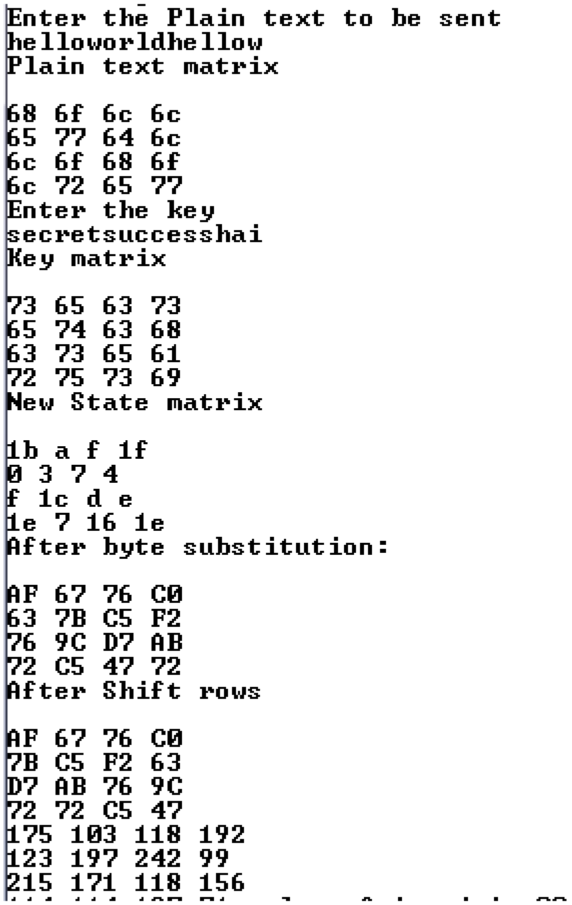
```
import java.util.*;
import java.io.*;

public class SAES {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the Plain text to be sent");
        String pt = sc.nextLine();
        String p[][] = new String[4][4];
        int k = 0;
        System.out.println("Plain text matrix");
        for (int i = 0; i < 4; i++) {
            //System.out.println(i);
            for (int j = 0; j < 4; j++) {
                //  System.out.println(j);
                int t = pt.charAt(k);
                String te = Integer.toHexString(t);
                p[j][i] = te;
                k++;
            }
        }
        for (int i = 0; i < 4; i++) {
            System.out.println();
            for (int j = 0; j < 4; j++) {
                System.out.print(p[i][j] + " ");
            }
        }
        System.out.println();
        System.out.println("Enter the key");
        String s[][] = new String[4][4];
        String key = sc.nextLine();
        k = 0;
        System.out.println("Key matrix");
        for (int i = 0; i < 4; i++) {
            //System.out.println(i);
            for (int j = 0; j < 4; j++) {
                //  System.out.println(j);
                int t = key.charAt(k);
                String te = Integer.toHexString(t);
                s[j][i] = te;
                k++;
            }
        }
        for (int i = 0; i < 4; i++) {
            System.out.println();
            for (int j = 0; j < 4; j++) {
                System.out.print(s[i][j] + " ");
            }
        }
        //exor key and pt
        System.out.println();
        System.out.println("New State matrix");
        String s1[][] = new String[4][4];
```

```
        for (int i = 0; i < 4; i++) {
            for (int j = 0; j < 4; j++) {
                int ss = Integer.parseInt(s[i][j], 16);
                int ps = Integer.parseInt(p[i][j], 16);
                int xor = ss ^ ps;

                s1[i][j] = Integer.toHexString(xor);
                // System.out.print(s1[i][j]+" ");
            }
        }
        for (int i = 0; i < 4; i++) {
            System.out.println();
            for (int j = 0; j < 4; j++) {
                System.out.print(s1[i][j] + " ");
            }
        }
        String sbox[][] = {{"63", "7C", "77", "7B", "F2", "6B", "6F", "C5", "30",
"01", "67", "2B", "FE", "D7", "AB", "76"}, {"CA", "82", "C9", "7D", "FA", "59",
"47", "F0", "AD", "D4", "A2", "AF", "9C", "A4", "72", "C0"}, {"B7", "FD",
"93", "26", "36", "3F", "F7", "CC", "34", "A5", "E5", "F1", "71", "D8", "31",
"15"}, {"04", "C7", "23", "C3", "18", "96", "05", "9A", "07", "12", "80", "E2",
"EB", "27", "B2", "75"}, {"09", "83", "2C", "1A", "1B", "6E", "5A", "A0", "52",
"3B", "D6", "B3", "29", "E3", "2F", "84"}, {"53", "D1", "00", "ED", "20", "FC",
"B1", "5B", "6A", "CB", "BE", "39", "4A", "4C", " 58", "CF"}, {"D0", "EF",
"AA", "FB", "43", "4D", "33", "85", "45", "F9", "02", "7F", "50", "3C", "9F",
"A8"}, {"51", "A3", "40", "8F", "92", "9D", "38", "F5", "BC", "B6", "DA", "21",
"10", "FF", "F3", "D2"}, {"CD", "0C", "3", "EC", "5F", "97", "44", "7", "C4",
"A7", "7E", "3D", "64", "5D", "19", "73"}, {"60", "81", "4F", "DC", "22", "2A",
"90", "88", "46", "EE", "B8", "14", "DE", "5E", "0B", "DB"}, {"E0", "32", "3A",
"0A", "49", "06", "24", "5C", "C2", "D3", "AC", "62", "91", "95", "E4", "79"},
{"E7", "C8", "37", "6D", "8D", "D5", "4E", "A9", "6C", "56", "F4", "EA", "65",
"7A", "AE", "08"}, {"BA", "78", "25", "2E", "1C", "A6", "B4", "C6", "E8",
"DD", "74", "1F", "4B", "BD", "8B", "8A"}, {"70", "3E", "B5", "66", "48", "03",
"F6", "0E", "61", "35", "57", "B9", "86", "C1", "1D", "9E"}, {"E1", "F8", "98",
"11", "69", "D9", "8E", "94", "9B", "1E", "87", "E9", "CE", "55", "28", "DF"},
{"8C", "A1", "89", "0D", "BF", "E6", "42", "68", "41", "99", "2D", "0F", "B0",
"54", "BB", "16"}};
        String bs[][] = new String[4][4];
        //byte substitution
        for (int i = 0; i < 4; i++) {
            for (int j = 0; j < 4; j++) {
                int r, c;
                if (s1[i][j].length() < 2) {
                    r = 0;

                    if (s1[i][j].equals("a") || s1[i][j].equals("b") || s1[i][j].equals("c") ||
s1[i][j].equals("d") || s1[i][j].equals("e") || s1[i][j].equals("f")) {
                        c = s1[i][j].charAt(0) - 97 + 10;
                    } else {
                        c = s1[i][j].charAt(0) - 48;
                    }
                } else {

                    if (s1[i][j].charAt(0) == 'a' || s1[i][j].charAt(0) == 'b' ||
s1[i][j].charAt(0) == 'c' || s1[i][j].charAt(0) == 'd' || s1[i][j].charAt(0) == 'e' ||
s1[i][j].charAt(0) == 'f') {
                        r = s1[i][j].charAt(0) - 97 + 10;
                    } else {
                        r = s1[i][j].charAt(0) - 48;
                    }
                    if (s1[i][j].charAt(1) == 'a' || s1[i][j].charAt(1) == 'b' ||
s1[i][j].charAt(1) == 'c' || s1[i][j].charAt(1) == 'd' || s1[i][j].charAt(1) == 'e' ||
s1[i][j].charAt(1) == 'f') {
                        c = s1[i][j].charAt(1) - 97 + 10;
                    } else {
                        c = s1[i][j].charAt(1) - 48;
                    }
                }
                bs[i][j] = sbox[r][c];
            }
        }
```

```
System.out.println();
System.out.println("After byte substitution:");
for (int i = 0; i < 4; i++) {
    System.out.println();
    for (int j = 0; j < 4; j++) {
        System.out.print(bs[i][j] + " ");
    }
}
//shift rows
String t;
t = bs[1][0];
for (int i = 0; i < 3; i++) {
    bs[1][i] = bs[1][i + 1];
}
bs[1][3] = t;
String temp;
temp = bs[2][0];
t = bs[2][1];
bs[2][0] = bs[2][2];
bs[2][1] = bs[2][3];
bs[2][2] = temp;
bs[2][3] = t;

temp = bs[3][0];
t = bs[3][1];
String te;
te = bs[3][2];
bs[3][0] = bs[3][3];
bs[3][1] = temp;
bs[3][2] = t;
bs[3][3] = te;
System.out.println();
System.out.println("After Shift rows");
for (int i = 0; i < 4; i++) {
    System.out.println();
    for (int j = 0; j < 4; j++) {
```

```
                System.out.print(bs[i][j] + " ");
            }
        }
        //mixcolumns

        int mix[][] = {{2, 3, 1, 1}, {1, 2, 3, 1}, {1, 1, 2, 3}, {3, 1, 1, 2}};
        int mc[][] = new int[4][4];
        for (int i = 0; i < 4; i++) {
            for (int j = 0; j < 4; j++) {
                mc[i][j] = Integer.parseInt(bs[i][j], 16);
            }
        }
        for (int i = 0; i < 4; i++) {
            System.out.println();
            for (int j = 0; j < 4; j++) {
                System.out.print(mc[i][j] + " ");
            }
        }
        int y = 283;
        int r[][] = new int[4][4];
        int x;
        for (int i = 0; i < 4; i++) {
            for (int j = 0; j < 4; j++) {
                r[i][j] = 0;
                for (int ki = 0; ki < 4; ki++) {
                    System.out.println("value of i n j is " + i + j);
                    x = mc[ki][j];
                    switch (mix[i][ki]) {

                        case 2:

                            // System.out.println("value of mc is " + mc[ki][j] + " value of
mix is " + mix[i][ki]);
                            //x = x << 1;
                            if (mc[ki][j] % 2 != 0) {
                                x = (x << 1) ^ y;
                                //System.out.println("Before Xor with 11b x=" + x);
                                //  x = x ^ y;
                                //System.out.println("After Xor with 11b x=" + x);
                            } else {
                                x = x << 1;
                            }
                            break;
                        case 3:
                            // System.out.println("value of mc is " + mc[ki][j] + " value of
mix is " + mix[i][ki]);

                            // x = x << 1;
                            if (mc[ki][j] % 2 != 0) {
                                x = ((x << 1) ^ y) ^ x;
                                //   System.out.println("Before Xor with 11b x=" + x);
                                //x = x ^ y;
                                // System.out.println("After Xor with 11b x=" + x);
                            } else {
                                x = (x << 1) ^ x;
                            }
                            //x = x ^ mc[ki][j];
                            break;
                        default:
                            break;
                    }
                    r[i][j] = r[i][j] ^ x;
                }
            }
        }
        for (int i = 0; i < 4; i++) {
            System.out.println();
            for (int j = 0; j < 4; j++) {
                System.out.print(r[i][j] + " ");
            }
        }
        for (int i = 0; i < 4; i++) {
            for (int j = 0; j < 4; j++) {
                bs[i][j] = Integer.toHexString(r[i][j]);
            }
        }
        System.out.println();
        System.out.println("After mix columns");
        for (int i = 0; i < 4; i++) {
            System.out.println();
            for (int j = 0; j < 4; j++) {
                System.out.print(bs[i][j] + " ");
            }
        }
    }

}
```

Output :



```
Enter the Plain text to be sent
helloworldhellow
Plain text matrix

68 6f 6c 6c
65 77 64 6c
6c 6f 68 6f
6c 72 65 77
Enter the key
secretsuccesshai
Key matrix

73 65 63 73
65 74 63 68
63 73 65 61
72 75 73 69
New State matrix

1b a f 1f
0 3 7 4
f 1c d e
1e 7 16 1e
After byte substitution:

AF 67 76 C0
63 7B C5 F2
76 9C D7 AB
72 C5 47 72
After Shift rows

AF 67 76 C0
7B C5 F2 63
D7 AB 76 9C
72 72 C5 47
175 103 118 192
123 197 242 99
215 171 118 156
```

RESULT:

Thus the advanced encryption standard algorithm has been implemented successfully.

Ex.No:6

RC4 ALGORITHM

AIM:

To write a program to implement rc4 algorithm for encryption and decryption
.

ALGORITHM:

- Start the program
- Get the input plaintext and key from the user.
- Initialize s[i] as i inside a loop ranging from 0 to 255 and compute t[i].
- T[i]=k[I mod keylen]
- Now use t to produce the initial permutation of s.let j=0
- J=(j+s[i]+t[i])mod 256
- Swap (s[i],t[i])repeat this for all the 256 values of i.
- Generate the stream to get k value for encryption and decryption.
- Stop the program.

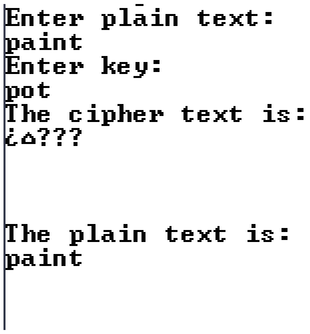
SOURCE CODE:

```
import java.util.*;

public class rc4 {

    static Scanner s = new Scanner(System.in);

    public static void main(String[] args) {
        int S[] = new int[256], i, t, k;
        String K;
        int kl, j = 0, temp;
        int T[] = new int[256];
        char ct[] = new char[256];
        String p;
        int pl;
        char pt[] = new char[256];
        System.out.println("Enter plain text:");
```

```
p = s.next();
pl = p.length();
System.out.println("Enter key:");
K = s.next();
kl = K.length();
for (i = 0; i < 256; i++) {
    S[i] = i;
    T[i] = (int) K.charAt(i % kl) - 97;
}
for (i = 0; i < 256; i++) {
    j = (j + S[i] + T[i]) % 256;
    temp = S[i];
    S[i] = S[j];
    S[j] = temp;
}
i = 0;
j = 0;
int m = 0, n = 0;
while (pl != 0) {
    i = (i + 1) % 256;
    j = (j + S[i]) % 256;
    temp = S[i];
    S[i] = S[j];
    S[j] = temp;
    t = (S[i] + S[j]) % 256;
    k = S[t];
    ct[m] = (char) ((((int) p.charAt(m) - 97) ^ k) + 97);
    pt[n] = (char) ((((int) ct[m] - 97) ^ k) + 97);
    m++;
    n++;
    pl--;
}
System.out.println("The cipher text is:");
System.out.println(ct);
System.out.println("The plain text is:");
System.out.println(pt);
```

```
    }


}
```

Output :



```
Enter plain text:
paint
Enter key:
pot
The cipher text is:
¿⌂???


The plain text is:
paint
```

RESULT:

Thus the rc4 algorithm has been implemented successfully.

Ex.No:7

DIFFIE HELLMAN

AIM:

To write a program to implement key exchange between two users using diffie hellman algorithm.

ALGORITHM:

1.Start the program

2.Get the prime number ‘q’ and calculate its primitive roots and choose least primitive root as alpha.

3.Get the public component xa and xb from the user.

4.Calculate the private components ya and yb .

5.Stop the program.

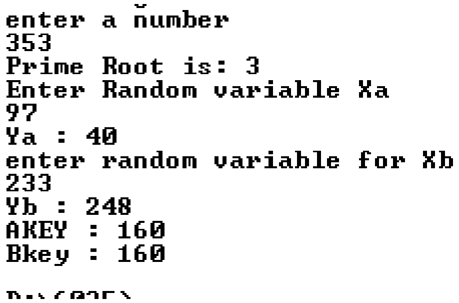
Source code:

```
import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;
import java.lang.*;
import static java.lang.Math.pow;
import static java.lang.Math.sqrt;
import java.math.BigInteger;
import java.util.HashMap;
import java.util.HashSet;
import java.util.Map;
import java.util.Set;
public class Deffiehelman {
public void checkprime(int n){
        int i,m=0,flag=0;
                m=n/2;
            if(n==0||n==1){
             System.out.println(n+" is not prime number");
            }else{
             for(i=2;i<=m;i++){
              if(n%i==0){
               System.out.println(n+" is not prime number");
               flag=1;
               break;
              }
             }}
    }
    public int primitiveroot(int n){
        int m = n;
        m = n - 1;
        int primeRoot = 0;
        Map<Integer, Integer> primeFactor = getPrimeFactor(m);
        for (Map.Entry<Integer, Integer> map : primeFactor.entrySet()) {
            primeFactor.put(map.getKey(), m / map.getKey());
        }
        for (int i = 2; i <= m; i++) {
            boolean notPrimeRoot = false;
            Set<Integer> reminder = new HashSet<>();
            for (Map.Entry<Integer, Integer> map : primeFactor.entrySet()) {
                if(BigInteger.valueOf(i).modPow(BigInteger.valueOf(map.getValue()),
BigInteger.valueOf(n)).equals(BigInteger.ONE))
                    notPrimeRoot = true;
            }
            if (!notPrimeRoot) {
                primeRoot = i;
                break;
            }
        }

        System.out.println("Prime Root is: " + primeRoot);
        return primeRoot;
    }


    private static Map<Integer, Integer> getPrimeFactor(int p) {
        Map<Integer, Integer> map = new HashMap<>();
        while (p % 2 == 0) {
            insertToMap(2, map);
            p /= 2;
        }

        for (int i = 3; i <= Math.sqrt(p); i += 2) {
            while (p % i == 0) {
                insertToMap(i, map);
                p /= i;
            }
        }

        if (p > 2)
            insertToMap(p, map);
        return map;
    }

    private static void insertToMap(int i, Map<Integer, Integer> map) {
        if (map.get(i) != null) {
            map.put(i, map.get(i) + 1);
        } else {
            map.put(i, 1);
        }
    }
    public int userA(int n,int primeRoot,int Xa){
        int res = 1;
        primeRoot = primeRoot % n;

        while (Xa > 0)
        {
            if((Xa & 1)==1)
                res = (res * primeRoot) % n;
            Xa = Xa >> 1;
            primeRoot = (primeRoot * primeRoot) % n;
        }
        return res;

    }
```

```
public int userB(int n,int primeRoot,int Xb){
    int res = 1;
     primeRoot = primeRoot % n;
    while (Xb > 0)
    {
        if((Xb & 1)==1)
            res = (res * primeRoot) % n;
        Xb = Xb >> 1;
        primeRoot = (primeRoot * primeRoot) % n;
    }
    return res;


}
public int Akey(int YB,int Xa,int n){
    int res = 1;
    YB = YB % n;
    while (Xa > 0)
    {
        if((Xa & 1)==1)
            res = (res * YB) % n;
        Xa = Xa >> 1;
        YB = (YB * YB) % n;
    }
    return res;

}

public int Bkey(int YA,int Xb,int n){
    int res = 1;
    YA = YA % n;
    while (Xb > 0)
    {
        if((Xb & 1)==1)
            res = (res * YA) % n;
        Xb = Xb >> 1;
```

```
            YA = (YA * YA) % n;
         }
        return res;

    }

    public static void main(String[] args) {
        Deffiehelman dh=new Deffiehelman();
        Scanner s=new Scanner(System.in);
        System.out.println("enter a number");
        int n=s.nextInt();
        dh.checkprime(n);
        int primeRoot= dh.primitiveroot(n);
        System.out.println("Enter Random variable Xa");
        int Xa=s.nextInt();
        int YA=  dh.userA(n,primeRoot , Xa);
        System.out.println("Ya : "+YA);
        System.out.println("enter random variable for Xb");
        int Xb=s.nextInt();
        int YB=dh.userB(n, primeRoot, Xb);
        System.out.println("Yb : "+YB);
        int Ak=dh.Akey(YB, Xa, n);
        System.out.println("AKEY : "+Ak);
        int Bk=dh.Bkey(YA, Xb, n);
        System.out.println("Bkey : "+Bk);}}
```

Output:



```
enter a number
353
Prime Root is: 3
Enter Random variable Xa
97
Ya : 40
enter random variable for Xb
233
Yb : 248
AKEY : 160
Bkey : 160
```

RESULT: thus the diffie hellman program implemented successfully.

Ex.No:8

RIVEST SHAMIR ADLEMAN(RSA)

AIM:

To implement rsa algorithm for encryption and decryption.

ALGORITHM:

1.Start the program
2.Get the inputs such as p,q(prime numbers) a random exponent e and message m from the user.
3.Find the secretkey component.
4.For encryption use public key pair(e,n)
5.For decryption private key pair(d,n)
6.Print the encrypted and decrypted text .
7.Stop the program.

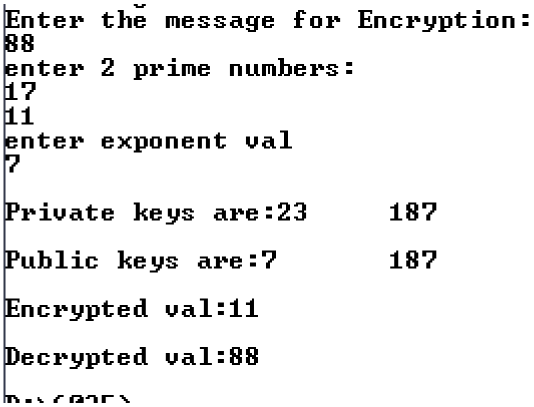
SOURCE CODE:

```
import java.io.*;
import java.util.*;
public class Rsa
{
    public static int encrypt(int x, int y, int p)
    {
      int res = 1;
   x = x % p;
while (y > 0
if((y & 1)==1)
res = (res * x) % p;
y >> 1;
X=(x * x) % p;
}

return res;
    }
public static int modInverse(int a, int m)

{
```

```
a = a % m;
for (int x = 1; x < m; x++)
if ((a * x) % m == 1)
return x;
return 1;
}
 public static int decrypt(int x, int y, int p)
    {
           int res = 1;
          x = x % p;
while (y > 0)
{
if((y & 1)==1)
res = (res * x) % p;
y = y >> 1;
x = (x * x) % p;
}
return res;
    }

    public static void main(String[] args)
    {

    Scanner sc = new Scanner(System.in);
        System.out.println("Enter the message for Encryption: ");
 int message = sc.nextInt();
        System.out.println("enter 2 prime numbers: ");
        int p=sc.nextInt();
        int q=sc.nextInt();
        int n=p*q;
        int pn=(p-1)*(q-1);
         System.out.println("enter exponent val");
                                                 int e=sc.nextInt();
         int d=modInverse(e,pn);

         System.out.println("\nPrivate keys are:"+d +"\t"+n);
```

```
                        System.out.println("\nPublic keys are:"+e +"\t"+n);

        System.out.println("\nEncrypted val:"+encrypt(message, e, n));
        System.out.println("\nDecrypted val:"+decrypt(encrypt(message, e, n), d,n));

  }}
```

Output:



```
Enter the message for Encryption:
88
enter 2 prime numbers:
17
11
enter exponent val
7

Private keys are:23     187

Public keys are:7       187

Encrypted val:11

Decrypted val:88

D:\(03E\
```

RESULT:

Thus the RSA algorithm has been implemented successfully.

Ex.No:9

ELGAMMAL SIGNATURE

AIM:

To implement elgammal signature for encryption and decryption .

ALGORITHM:

- Start the program
- Get the input values from the user.
- Get the value of primitive root.
- Generate public key and private key component.
- Choose the hash value from 0 to q-1.
- And choose another k value from 1 to q-1.
- Finally gives the signature pair(p,q) and also gives the vertex.
- Stop the program.

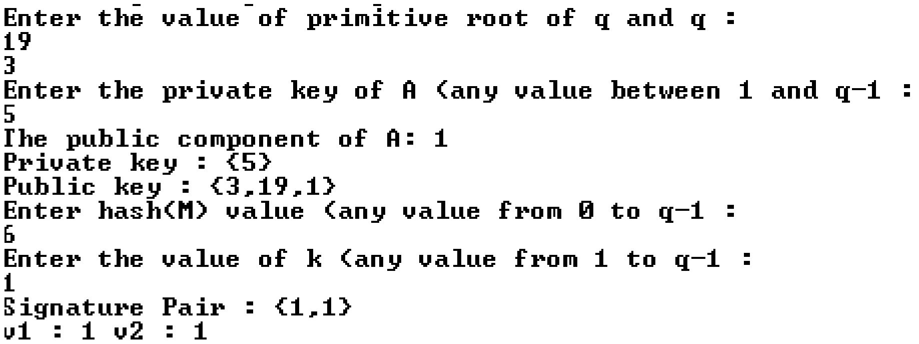
SOURCE CODE:

```
import java.util.*;
/**
 *
 * @author student
 */
public class Elgammalsignature {
static Scanner s = new Scanner(System.in);
static int Q = 0,mi=0;
    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        int m,q,a,x_a,y_a,k,s1,s2,k_inv,t,v1,v2,t1;
        System.out.println("Enter the value of primitive root of q and q : ");
        a=s.nextInt();
        q=s.nextInt();
        System.out.println("Enter the private key of A (any value between 1 and q-1
: ");
```

```
        x_a=s.nextInt();
        y_a=mod(a,x_a,q);
        System.out.println("The public component of A: "+y_a);
        System.out.println("Private key : {"+x_a+"}");
        System.out.println("Public key : {"+q+","+a+","+y_a+"}");
        System.out.println("Enter hash(M) value (any value from 0 to q-1 : ");
        m=s.nextInt();
        System.out.println("Enter the value of k (any value from 1 to q-1 : ");
        k=s.nextInt();
        s1=mod(a,k,q);
        k_inv=inv(1,0,(q-1),0,1,k,0,0,0);
        t=k_inv*(m-(x_a*s1));
        s2=mod(t,1,(q-1));
        System.out.println("Signature Pair : {"+s1+","+s2+"}");
        v1=mod(a,m,q);
        t1= ((int)Math.pow(y_a,s1))*((int)Math.pow(s1,s2));
        v2=mod(t1,1,q);
        System.out.println("v1 : "+v1+" v2 : "+v2);
    }
    public static int mod(int a,int p,int n)
    {
        int x=1;
        int j=a;
         while (p > 0) {
            if (p % 2 == 1) {
                x = (x * a);
                if (x > n) {
                    x %= n;
                }
            }
            a = (a * a);
            if (a > n) {
                a %= n;
            }
            p /= 2;
        }
```

```
          return x;
     }
public static int inv(int A1,int A2,int A3,int B1,int B2,int B3,int T1,int T2,int T3)
{

if(B3==1) {
mi=B2;

}
else{
    Q = Math.abs(A3/B3);
update(A1,A2,A3,B1,B2,B3,T1,T2,T3,Q);}
return mi;
}
public static void update(int A1,int A2,int A3,int B1,int B2,int B3,int T1,int
T2,int T3,int Q)
{
T1=A1-Q*B1;
T2=A2-Q*B2;
T3=A3-Q*B3;
A1=B1;A2=B2; A3=B3; B1=T1; B2=T2; B3=T3;
inv(A1,A2,A3,B1,B2,B3,T1,T2,T3);
}
}
```

OUTPUT:



```
Enter the value of primitive root of q and q :
19
3
Enter the private key of A (any value between 1 and q-1 :
5
Ihe public component of A: 1
Private key : {5}
Public key : {3,19,1}
Enter hash(M) value (any value from 0 to q-1 :
6
Enter the value of k (any value from 1 to q-1 :
1
Signature Pair : {1,1}
v1 : 1 v2 : 1
```

RESULT:

Thus the elgammal signature program has been implemented successfully.

Ex.No:10

SECURED HASH ALGORITHM(SHA)

AIM:

To implement secured hash algorithm to generate 128 bit hash value as the output.

ALGORITHM:

- Start the program.
- Initialize the buffers from a-h using the default.
- Initialize the values of wt and kt.
- Calculate ch(e,f,g)using formula.
- Finally calculate the t0 and t1 values update all values.
- Stop the program.

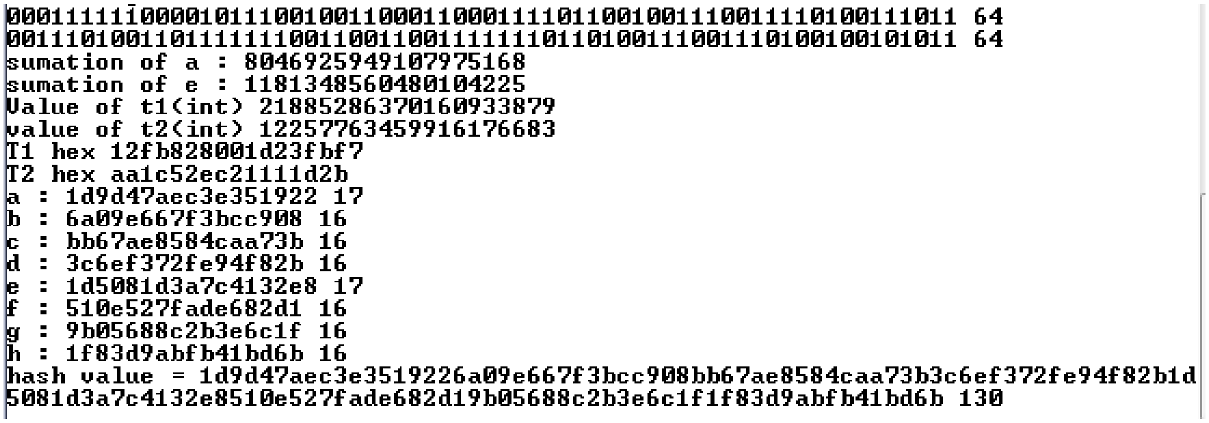
SOURCE CODE:

```
import java.math.BigInteger;
import java.util.*;

/**
 *
 * @author student
 */
public class SHA512 {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        String w = "6162638000000000";
        String k = "428a2f98d728ae22";
        String a = "6a09e667f3bcc908", e = "510e527fade682d1", b =
"bb67ae8584caa73b", f = "9b05688c2b3e6c1f", c = "3c6ef372fe94f82b", g =
"1f83d9abfb41bd6b", d =
"a54ff53a5f1d36f1", h = "5be0cd19137e2179";
        String ba = hexToBin(a);
        String bb = hexToBin(b);
        String bc = hexToBin(c);
        String bd = hexToBin(d);
        String be = hexToBin(e);
        String bf = hexToBin(f);
        String bg = hexToBin(g);
        String bh = hexToBin(h);
        BigInteger a1 = new BigInteger(be, 2);
        BigInteger a2 = new BigInteger(bf, 2);
        BigInteger a3 = new BigInteger(bg, 2);
        String ch = (a1.and(a2)).xor((a1.not()).and(a3)).toString(2);
        String p = "";
        int x = ch.length();
        // System.out.println(x);
        if (x < 64) {
            x = 64 - x;

            //System.out.println(x);
            for (int i = 0; i < x; i++) {
                p = p + "0";
            }

            //System.out.println(p);
            ch = p + ch;

            //System.out.println(ch);
        }
        BigInteger a4 = new BigInteger(ba, 2);
        BigInteger a5 = new BigInteger(bb, 2);
        BigInteger a6 = new BigInteger(bc, 2);
        String maj = ((a4.and(a5)).xor((a4.and(a6)).xor(a5.and(a6)))).toString(2);
        x = maj.length();

        //System.out.println(x);
```

```
if (x < 64) {
    x = 64 - x;

    //System.out.println(x);
    p = "";
    for (int i = 0; i < x; i++) {
        p = p + "0";
    }

    //System.out.println(p);
    maj = p + maj;

    //System.out.println(maj);
}
System.out.println(ch + " " + ch.length());
System.out.println(maj + " " + maj.length());
String r28 = Rotate(ba, 28);
String r34 = Rotate(ba, 34);
String r39 = Rotate(ba, 39);
BigInteger sa1 = new BigInteger(r28, 2);
BigInteger sa2 = new BigInteger(r34, 2);
BigInteger sa3 = new BigInteger(r39, 2);
String sa = ((sa1.xor(sa2)).xor(sa3)).toString();
String r14 = Rotate(be, 14);
String r18 = Rotate(be, 18);
String r41 = Rotate(be, 41);
BigInteger se1 = new BigInteger(r14, 2);
BigInteger se2 = new BigInteger(r18, 2);
BigInteger se3 = new BigInteger(r41, 2);
String se = ((se1.xor(se2)).xor(se3)).toString();
System.out.println("sumation of a : " + sa);
System.out.println("sumation of e : " + se);
BigInteger ta = new BigInteger(sa);
BigInteger te = new BigInteger(se);
BigInteger h1 = new BigInteger(h, 16);
BigInteger ch1 = new BigInteger(ch, 2);
BigInteger w1 = new BigInteger(w, 16);
BigInteger k1 = new BigInteger(k, 16);
BigInteger maj1 = new BigInteger(maj, 2);
BigInteger T1 = h1.add(ch1);
T1 = T1.add(te);
// System.out.println(T1);
T1 = T1.add(w1);
//System.out.println(T1);
T1 = T1.add(k1);
System.out.println("Value of t1(int) " + T1);
BigInteger T2 = ta.add(maj1);
System.out.println("value of t2(int) " + T2);
String t1 = T1.toString(16);
String t2 = T2.toString(16);
System.out.println("T1 hex " + t1);
System.out.println("T2 hex " + t2);
h = g;
g = f;
f = e;
BigInteger D = new BigInteger(d, 16);
e = (D.add(T1)).toString(16);
d = c;
c = b;
b = a;
a = (T1.add(T2)).toString(16);
System.out.println("a : " + a + " " + a.length());
System.out.println("b : " + b + " " + b.length());
System.out.println("c : " + c + " " + c.length());
System.out.println("d : " + d + " " + d.length());
System.out.println("e : " + e + " " + e.length());
System.out.println("f : " + f + " " + f.length());
System.out.println("g : " + g + " " + g.length());
System.out.println("h : " + h + " " + h.length());
String Final = a + b + c + d + e + f + g + h;
System.out.println("hash value = " + Final + " " + Final.length());
```

```
    }

    static String Rotate(String s, int n) {
        s += s;
        String r;
        r = s.substring(n, n + 64);
        return r;
    }

    static String hexToBin(String s) {
        String p = new BigInteger(s, 16).toString(2);
        String a = "";
        int x = p.length();
        if (x < 64) {
            x = 64 - x;

            for (int i = 0; i < x; i++) {
                a = a + "0";
            }
            a = a + p;
        } else {
            a = p;
        }
        return a;
    }
}
```

OUTPUT:



```
0001111110000101110010011000110001111011001001110011110100111011 64
0011101001101111111001100110011111110110100111001110100100101011 64
sumation of a : 8046925949107975168
sumation of e : 1181348560480104225
Value of t1(int) 2188528637016093387 9
value of t2(int) 1225776345991617668 3
T1 hex 12fb828001d23fbf7
T2 hex aa1c52ec21111d2b
a : 1d9d47aec3e351922 17
b : 6a09e667f3bcc908 16
c : bb67ae8584caa73b 16
d : 3c6ef372fe94f82b 16
e : 1d5081d3a7c4132e8 17
f : 510e527fade682d1 16
g : 9b05688c2b3e6c1f 16
h : 1f83d9abfb41bd6b 16
hash value = 1d9d47aec3e3519226a09e667f3bcc908bb67ae8584caa73b3c6ef372fe94f82b1d
5081d3a7c4132e8510e527fade682d19b05688c2b3e6c1f1f83d9abfb41bd6b 130
```

RESULT:

Thus the sha-512 program has been implemented successfully and output is displayed .

Ex.No:11

DIGITAL SIGNATURE STANDARD

AIM:

To implement dss algorithm and generate and verify the signatures generated.

ALGORITHM:

- Start the program
- input the default values.
- Choose a random value p and q.
- Calculate g value using formula
- Now choosen random value x.
- Now generate signature(r,s).
- Check whether r=v.
- Stop the program.

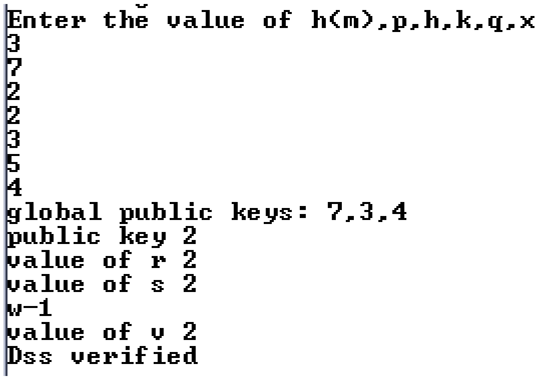
SOURCE CODE:

```
import java.io.*;
import java.util.*;
public class Dss
{
public static int inverse(int b,int m)
{
int q,a1,a2,a3,b1,b2,b3,t1,t2,t3;
a1=1;
a2=0;
a3=m;
b1=0;
b2=1;
b3=b;
while(b3!=1)
{
q=a3/b3;
t1=a1-(q*b1);
t2=a2-(q*b2);
t3=a3-(q*b3);
a1=b1;
a2=b2;
a3=b3;
b1=t1;
b2=t2;
b3=t3;
}
return b2;
}
public static void main(String args[])
{
Scanner sc=new Scanner(System.in);
System.out.println("Enter the value of h(m),p,h,k,q,x");
int hf=sc.nextInt();
int p=sc.nextInt();
int h=sc.nextInt();
int k=sc.nextInt();
int q=sc.nextInt();
int x=sc.nextInt();
int pow=(p-1)/q;
int g=1;
for(int i=0;i<pow;i++)
{
g=(g*h)%p;
}
System.out.println(g);
System.out.println("global public keys:"+" "+p+","+q+","+g);
int y=1;
for(int i=0;i<x;i++)
{
y=(y*g)%p;
}
System.out.println("public key"+" "+y);
int r1=1;
int r=0;
for(int i=0;i<k;i++)
```

```
{
r1=(r1*g)%p;
r=r1%q;
}
System.out.println("value of r"+" "+r);
int invert=inverse(k,q);
int t=x*r;
int s1=invert*(hf+t);
int st=-s1;
while(st>q)
st=st%q;
int s=q-st;
System.out.println("value of s"+" "+s);
int w=inverse(s,q);
System.out.println("w"+w);
int n1=(hf*w)%q;
//System.out.println("n1"+n1);
int n2=(r*w);
int st1=-n2;
if(st1>q){
st1=st1%q;}
int n21=q-st1;
//System.out.println("n2"+n21);
int v1=1;
for(int i=0;i<n1;i++)
{
v1=v1*g;
}
int v2=1;
for(int i=0;i<n21;i++)
{
v2=v2*y;
}
int v3=(v1*v2)%p;
int v=v3%q;
System.out.println("value of v"+" "+v);
```

```
if(r==v)
System.out.println("Dss verified");
else
System.out.println("Dss not verified");
}
}
```

OUTPUT:



```
Enter the value of h(m),p,h,k,q,x
3
7
2
2
3
5
4
global public keys: 7,3,4
public key 2
value of r 2
value of s 2
w-1
value of v 2
Dss verified
```

RESULT:

Thus the dss algorithm has been implemented successfully.

Ex.NO:12

GRAPHICAL PRIMITIVES AND ANIMATIONS IN ANDROID

AIM:

1. To write a program to implement graphical primitives in android.
2. To write a program to implement animation effects in android.

1. GRAPHICAL PRIMITIVES

DESCRIPTION/ ALGORITHM:

The android.graphics framework divides drawing into two areas:

- *What* to draw, handled by **Canvas**
- *How* to draw, handled by **Paint**.

For instance, Canvas provides a method to draw a line, while Paint provides methods to define that line's color. Canvas has a method to draw a rectangle, while Paint defines whether to fill that rectangle with a color or leave it empty. Simply put, Canvas defines shapes that you can draw on the screen, while Paint defines the color, style, font, and so forth of each shape you draw.

• The android.graphics.Canvas can be used to draw graphics in android. It provides methods to draw oval, rectangle, picture, text, line etc.

• The android.graphics.Paint class is used with canvas to draw objects. It holds the information of color and style.

PROGRAM (Displaying 2D graphics in android):

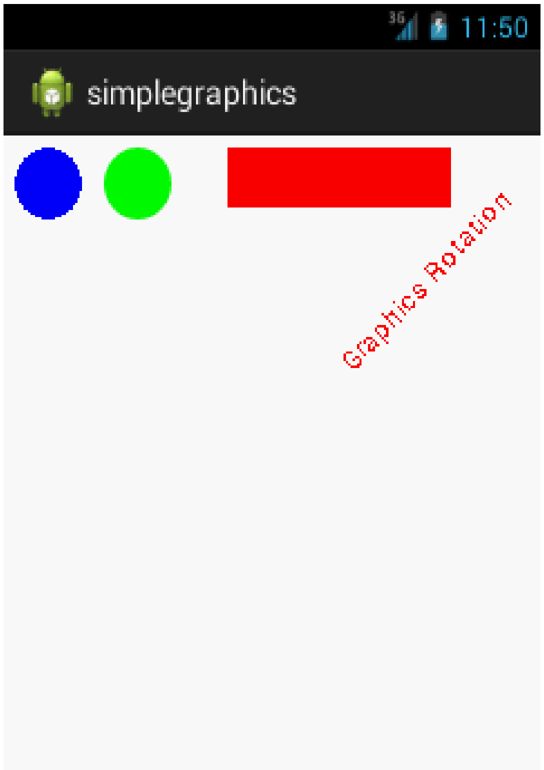
activity_main.xml

```
<RelativeLayout xmlns:androclass="http://schemas.android.com/apk/res/android"

    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:paddingBottom="@dimen/activity_vertical_margin"
    android:paddingLeft="@dimen/activity_horizontal_margin"
    android:paddingRight="@dimen/activity_horizontal_margin"
    android:paddingTop="@dimen/activity_vertical_margin"
    tools:context=".MainActivity" >
    <TextView
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/hello_world" />

</RelativeLayout>
```

MainActivity.java

```
package com.example.simplegraphics;

import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;
import android.content.Context;
import android.graphics.Canvas;
import android.graphics.Color;
import android.graphics.Paint;
import android.view.View;

public class MainActivity extends Activity {

    DemoView demoview;
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        demoview = new DemoView(this);
        setContentView(demoview);
    }

    private class DemoView extends View{
        public DemoView(Context context){
            super(context);
        }

        @Override protected void onDraw(Canvas canvas) {
            super.onDraw(canvas);

            // custom drawing code here
            Paint paint = new Paint();
        paint.setStyle(Paint.Style.FILL);

        // make the entire canvas white
        paint.setColor(Color.WHITE);
        canvas.drawPaint(paint);

        // draw blue circle with anti aliasing turned off
        paint.setAntiAlias(false);
        paint.setColor(Color.BLUE);
        canvas.drawCircle(20, 20, 15, paint);

        // draw green circle with anti aliasing turned on
        paint.setAntiAlias(true);
        paint.setColor(Color.GREEN);
        canvas.drawCircle(60, 20, 15, paint);

        // draw red rectangle with anti aliasing turned off
        paint.setAntiAlias(false);
        paint.setColor(Color.RED);
        canvas.drawRect(100, 5, 200, 30, paint);

        // draw the rotated text
        canvas.rotate(-45);

        paint.setStyle(Paint.Style.FILL);
        canvas.drawText("Graphics Rotation", 40, 180, paint);

        //undo the rotate
        canvas.restore();
    }
}
@Override
public boolean onCreateOptionsMenu(Menu menu) {
    // Inflate the menu; this adds items to the action bar if it is present.
    getMenuInflater().inflate(R.menu.main, menu);
    return true;
}  }
```

OUTPUT:



2. ANIMATIONS

DESCRIPTION/ ALGORITHM:

Animation is the process of creating motion and shape change. Animation in android is possible from many ways. One of the widely used way of making animation is called tweened animation.

Tween Animation takes some parameters such as start value , end value, size , time duration , rotation angle e.t.c and perform the required animation on that object. It can be applied to any type of object. So in order to use this , android has provided us a class called Animation.

In order to perform animation in android , we are going to call a static function loadAnimation() of the class AnimationUtils. We are going to receive the result in an instance of Animation Object. In order to apply this animation to an object , we will just call the startAnimation() method of the object.

Step Description

1. You will use Android studio IDE to create an Android application and name it as My Application under a package name.
2. Modify src/MainActivity.java file to add animation code
3. Modify layout XML file res/layout/activity_main.xml add any GUI component if required.
4. Create a new folder under res directory and call it anim. Confim it by visiting res/anim
5. Right click on anim and click on new and select Android XML file. You have to create different files that are listed below.
6. Create files myanimation.xml, clockwise.xml, fade.xml, move.xml, blink.xml, slide.xml and add the XML code.
7. No need to change default string constants. Android studio takes care of default constants at values/string.xml.
8. Run the application and choose a running android device and install the application on it and verify the results.

PROGRAM:

MainActivity.java

```
package com.example.sample.myapplication;
import android.app.Activity;
import android.os.Bundle;
import android.view.View;
import android.view.animation.Animation;
import android.view.animation.AnimationUtils;
import android.widget.ImageView;
import android.widget.Toast;

public class MainActivity extends Activity {
   @Override
   protected void onCreate(Bundle savedInstanceState) {
      super.onCreate(savedInstanceState);
      setContentView(R.layout.activity_main);
   }

   public void clockwise(View view){
    ImageView image = (ImageView)findViewById(R.id.imageView);
    Animation animation = AnimationUtils.loadAnimation
(getApplicationContext(), R.anim.myanimation);
    image.startAnimation(animation);
  }

  public void zoom(View view){
    ImageView image = (ImageView)findViewById(R.id.imageView);
    Animation animation1 =
AnimationUtils.loadAnimation(getApplicationContext(),
      R.anim.clockwise);
    image.startAnimation(animation1);
  }

  public void fade(View view){
    ImageView image = (ImageView)findViewById(R.id.imageView);
    Animation animation1 =
      AnimationUtils.loadAnimation(getApplicationContext(),
        R.anim.fade);
    image.startAnimation(animation1);
  }

  public void blink(View view){
    ImageView image = (ImageView)findViewById(R.id.imageView);
    Animation animation1 =
      AnimationUtils.loadAnimation(getApplicationContext(),
        R.anim.blink);
    image.startAnimation(animation1);
  }

  public void move(View view){
    ImageView image = (ImageView)findViewById(R.id.imageView);
    Animation animation1 =
      AnimationUtils.loadAnimation(getApplicationContext(), R.anim.move);
    image.startAnimation(animation1);
  }

  public void slide(View view){
    ImageView image = (ImageView)findViewById(R.id.imageView);
    Animation animation1 =
      AnimationUtils.loadAnimation(getApplicationContext(), R.anim.slide);
      image.startAnimation(animation1);
   }
}
```

activity_main.xml

```
<RelativeLayout
   xmlns:android="http://schemas.android.com/apk/res/android"
   xmlns:tools="http://schemas.android.com/tools"
   android:layout_width="match_parent"
   android:layout_height="match_parent"
   android:paddingLeft="@dimen/activity_horizontal_margin"
   android:paddingRight="@dimen/activity_horizontal_margin"
   android:paddingTop="@dimen/activity_vertical_margin"
   android:paddingBottom="@dimen/activity_vertical_margin"
tools:context=".MainActivity">

   <TextView
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Alert Dialog"
      android:id="@+id/textView"
      android:textSize="35dp"
      android:layout_alignParentTop="true"
      android:layout_centerHorizontal="true" />

   <TextView
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="Tutorialspoint"
      android:id="@+id/textView2"
      android:textColor="#ff3eff0f"
      android:textSize="35dp"
      android:layout_below="@+id/textView"
      android:layout_centerHorizontal="true" />

   <ImageView
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:id="@+id/imageView"
      android:src="@drawable/abc"
      android:layout_below="@+id/textView2"
      android:layout_alignRight="@+id/textView2"
      android:layout_alignEnd="@+id/textView2"
      android:layout_alignLeft="@+id/textView"
      android:layout_alignStart="@+id/textView"/>

   <Button
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="zoom"
      android:id="@+id/button"
      android:layout_below="@+id/imageView"
      android:layout_alignParentLeft="true"
      android:layout_alignParentStart="true"
      android:layout_marginTop="40dp"
      android:onClick="clockwise"/>

   <Button
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="clockwise"
      android:id="@+id/button2"
      android:layout_alignTop="@+id/button"
      android:layout_centerHorizontal="true"
      android:onClick="zoom"/>

   <Button
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="fade"
      android:id="@+id/button3"
      android:layout_alignTop="@+id/button2"
      android:layout_alignParentRight="true"
      android:layout_alignParentEnd="true"
      android:onClick="fade"/>

   <Button
      android:layout_width="wrap_content"
      android:layout_height="wrap_content"
      android:text="blink"
      android:onClick="blink"
      android:id="@+id/button4"
```

```
    android:layout_below="@+id/button"
    android:layout_alignParentLeft="true"
    android:layout_alignParentStart="true" />

  <Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="move"
    android:onClick="move"
    android:id="@+id/button5"
    android:layout_below="@+id/button2"
    android:layout_alignRight="@+id/button2"
    android:layout_alignEnd="@+id/button2"
    android:layout_alignLeft="@+id/button2"
    android:layout_alignStart="@+id/button2" />

  <Button
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="slide"
    android:onClick="slide"
    android:id="@+id/button6"
    android:layout_below="@+id/button3"
    android:layout_toRightOf="@+id/textView"
    android:layout_toEndOf="@+id/textView" />

</RelativeLayout>
```

res/anim/myanimation.xml

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android">

  <scale xmlns:android="http://schemas.android.com/apk/res/android"
    android:fromXScale="0.5"
    android:toXScale="3.0"
    android:fromYScale="0.5"
    android:toYScale="3.0"
    android:duration="5000"
    android:pivotX="50%"
    android:pivotY="50%" >
  </scale>

  <scale xmlns:android="http://schemas.android.com/apk/res/android"
    android:startOffset="5000"
    android:fromXScale="3.0"
    android:toXScale="0.5"
    android:fromYScale="3.0"
    android:toYScale="0.5"
    android:duration="5000"
    android:pivotX="50%"
    android:pivotY="50%" >
  </scale>
</set>
```

res/anim/clockwise.xml

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android">

  <rotate xmlns:android="http://schemas.android.com/apk/res/android"
    android:fromDegrees="0"
    android:toDegrees="360"
    android:pivotX="50%"
    android:pivotY="50%"
    android:duration="5000" >
  </rotate>

  <rotate xmlns:android="http://schemas.android.com/apk/res/android"
    android:startOffset="5000"
    android:fromDegrees="360"
    android:toDegrees="0"
    android:pivotX="50%"
    android:pivotY="50%"
    android:duration="5000" >
  </rotate> </set>
```

res/anim/fade.xml

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android"
  android:interpolator="@android:anim/accelerate_interpolator" >
  <alpha
    android:fromAlpha="0"
    android:toAlpha="1"
    android:duration="2000" >
  </alpha>

  <alpha
    android:startOffset="2000"
    android:fromAlpha="1"
    android:toAlpha="0"
    android:duration="2000" >
  </alpha>
</set>
```

res/anim/blink.xml

```
<?xml version="1.0" encoding="utf-8"?>
<set xmlns:android="http://schemas.android.com/apk/res/android">
  <alpha android:fromAlpha="0.0"
    android:toAlpha="1.0"
    android:interpolator="@android:anim/accelerate_interpolator"
    android:duration="600"
    android:repeatMode="reverse"
    android:repeatCount="infinite"/>
</set>
```

res/anim/move.xml

```
<?xml version="1.0" encoding="utf-8"?>
<set
  xmlns:android="http://schemas.android.com/apk/res/android"
  android:interpolator="@android:anim/linear_interpolator"
  android:fillAfter="true">

  <translate
    android:fromXDelta="0%p"
    android:toXDelta="75%p"
    android:duration="800" />
</set>
```

res/anim/slide.xml

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<set xmlns:android="http://schemas.android.com/apk/res/android"
    android:fillAfter="true" >

    <scale
        android:duration="500"
        android:fromXScale="1.0"
        android:fromYScale="1.0"
        android:interpolator="@android:anim/linear_interpolator"
        android:toXScale="1.0"
        android:toYScale="0.0" />
</set>
```

res/values/string.xml

```
<resources>
    <string name="app_name">My Application</string>
</resources>
```

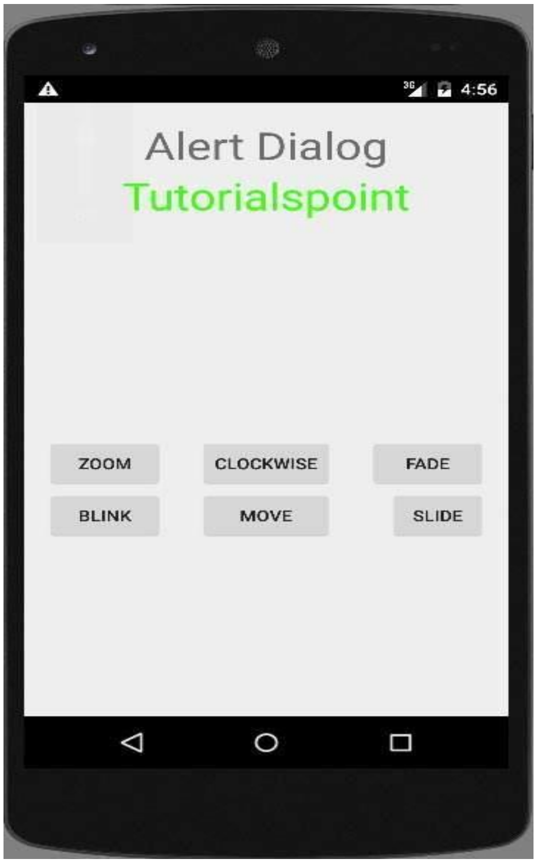
AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.sairamkrishna.myapplication" >

    <application
        android:allowBackup="true"
        android:icon="@drawable/ic_launcher"
        android:label="@string/app_name"
        android:theme="@style/AppTheme" >

        <activity
            android:name="com.example.animation.MainActivity"
            android:label="@string/app_name" >


             <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>

        </activity>
```

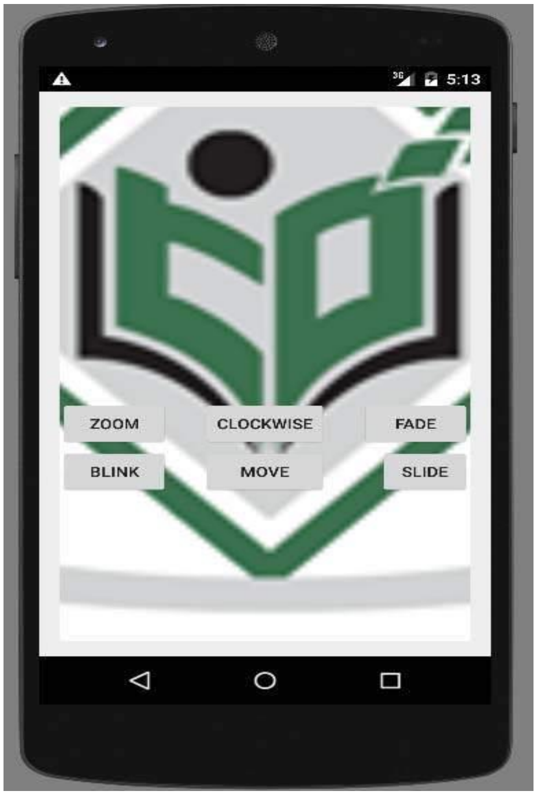
```
   </application>
</manifest>
```

OUTPUT:

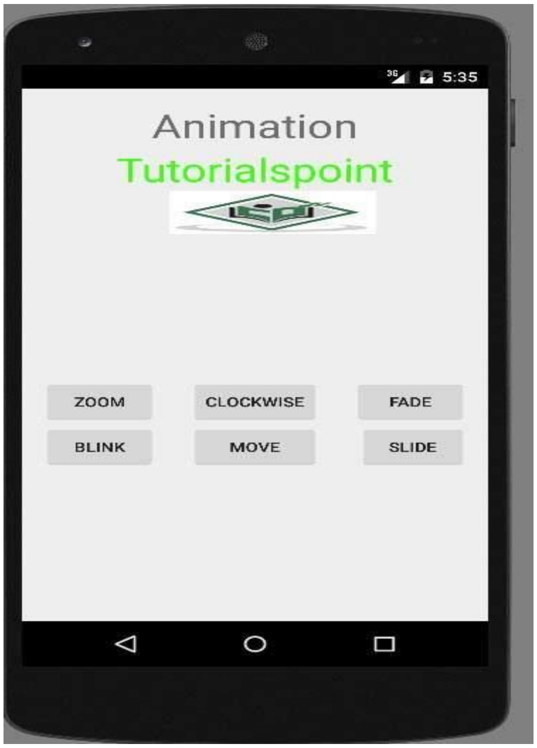
To run the app from Android studio, open one of your project's activity files and click Run ▶ icon from the toolbar. Android studio will display following images:



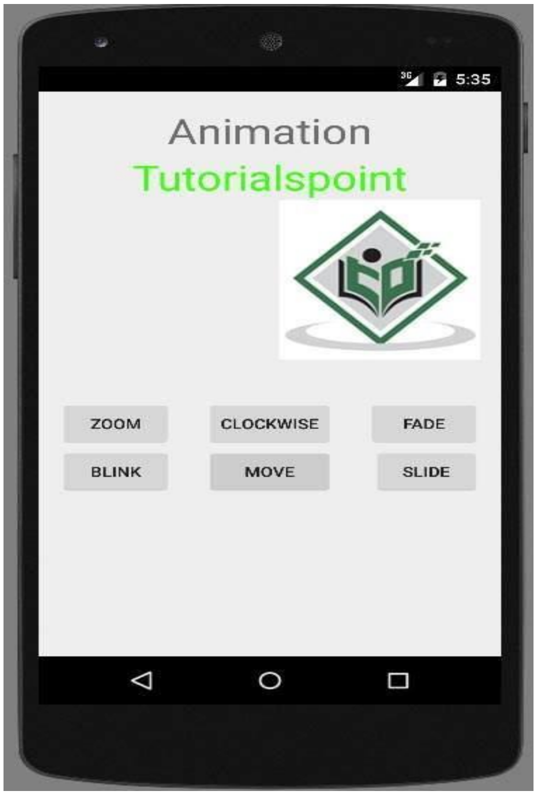
Select zoom button, it will display following screen –



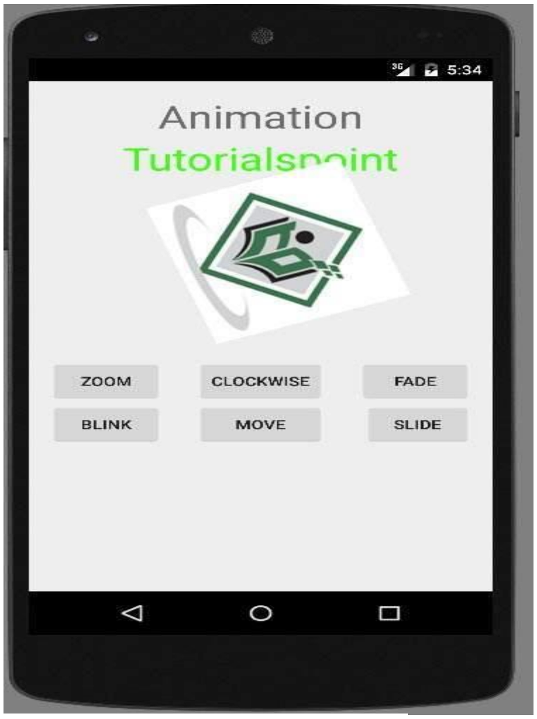
Now select slide button, it will display following screen-



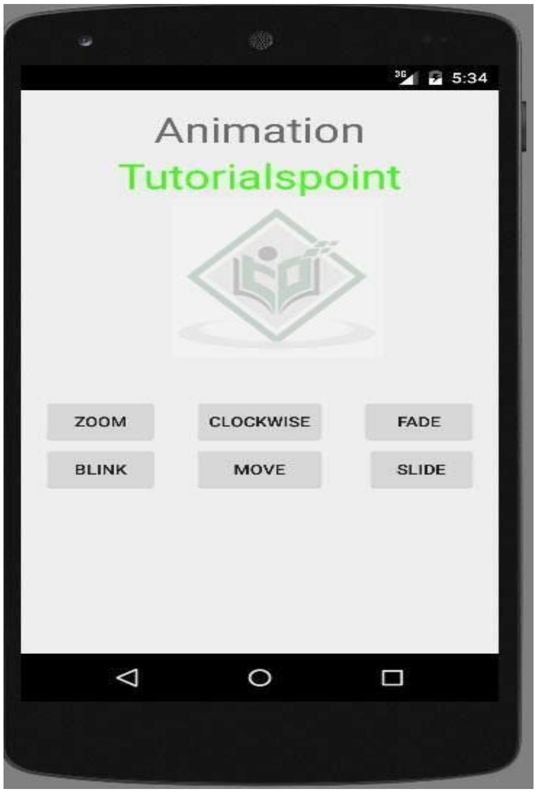
Now select move button, it will display following screen-



Now the clockwise button, it will display following screen-



Now Fade button, it will display following screen-



RESULT:

Thus, Graphical primitives and Animation effects are implemented in android successfully.

EX.NO:13

EVENT HANDLING

AIM:

To write a program to Implement Event Handling in android.

DESCRIPTON /ALGORITHM :

Events are a useful way to collect data about a user's interaction with interactive components of Applications. Like button presses or screen touch etc. The Android framework maintains an event queue as first-in, first-out (FIFO) basis. You can capture these events in your program and take appropriate action as per requirements.

There are following three concepts related to Android Event Management −

- **Event Listeners** − An event listener is an interface in the View class that contains a single callback method. These methods will be called by the Android framework when the View to which the listener has been registered is triggered by user interaction with the item in the UI.
- **Event Listeners Registration** − Event Registration is the process by which an Event Handler gets registered with an Event Listener so that the handler is called when the Event Listener fires the event.
- **Event Handlers** − When an event happens and we have registered an event listener for the event, the event listener calls the Event Handlers, which is the method that actually handles the event.

StepDescription

1 You will use Android studio IDE to create an Android application and name it as *myapplication* under a package *com.example.myapplication*

2 Modify *src/MainActivity.java* file to add click event listeners and handlers for the two buttons defined.

3 Modify the default content of *res/layout/activity_main.xml* file to include Android UI controls.

4 No need to declare default string constants.Android studio takes care

default constants.

5 Run the application to launch Android emulator and verify the result of the changes done in the application.

PROGRAM:

Following is the content of the modified main activity file src/com.example.myapplication/MainActivity.java. This file can include each of the fundamental lifecycle methods.

```
packagecom.example.myapplication;
importandroid.app.ProgressDialog;
importandroid.os.Bundle;
import android.support.v7.app.ActionBarActivity;
importandroid.view.View;
importandroid.widget.Button;
importandroid.widget.TextView;

public class MainActivity extends ActionBarActivity {
privateProgressDialog progress;
   Button b1,b2;

   @Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
progress = new ProgressDialog(this);

      b1=(Button)findViewById(R.id.button);
      b2=(Button)findViewById(R.id.button2);
b1.setOnClickListener(new View.OnClickListener() {

         @Override
public void onClick(View v) {
TextViewtxtView = (TextView) findViewById(R.id.textView);
txtView.setTextSize(25);
         }
```

```
     });

b2.setOnClickListener(new View.OnClickListener() {

       @Override
public void onClick(View v) {
TextViewtxtView = (TextView) findViewById(R.id.textView);
txtView.setTextSize(55);
        }
     });
   }
}
```

Following will be the content of res/layout/activity_main.xml file −

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:paddingBottom="@dimen/activity_vertical_margin"
android:paddingLeft="@dimen/activity_horizontal_margin"
android:paddingRight="@dimen/activity_horizontal_margin"
android:paddingTop="@dimen/activity_vertical_margin"
tools:context=".MainActivity">

<TextView
android:id="@+id/textView1"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Event Handling "
android:layout_alignParentTop="true"
android:layout_centerHorizontal="true"
android:textSize="30dp"/>

<TextView
```

```
android:id="@+id/textView2"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Tutorials point "
android:textColor="#ff87ff09"
android:textSize="30dp"
android:layout_above="@+id/imageButton"
android:layout_centerHorizontal="true"
android:layout_marginBottom="40dp" />

<ImageButton
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:id="@+id/imageButton"
android:src="@drawable/abc"
android:layout_centerVertical="true"
android:layout_centerHorizontal="true" />

<Button
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Small font"
android:id="@+id/button"
android:layout_below="@+id/imageButton"
android:layout_centerHorizontal="true" />

<Button
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Large Font"
android:id="@+id/button2"
android:layout_below="@+id/button"
android:layout_alignRight="@+id/button"
android:layout_alignEnd="@+id/button" />

<TextView
android:layout_width="wrap_content"
```

```
android:layout_height="wrap_content"
android:text="Hello World!"
android:id="@+id/textView"
android:layout_below="@+id/button2"
android:layout_centerHorizontal="true"
android:textSize="25dp" />

</RelativeLayout>
```

Following will be the content of res/values/strings.xml to define two new constants −

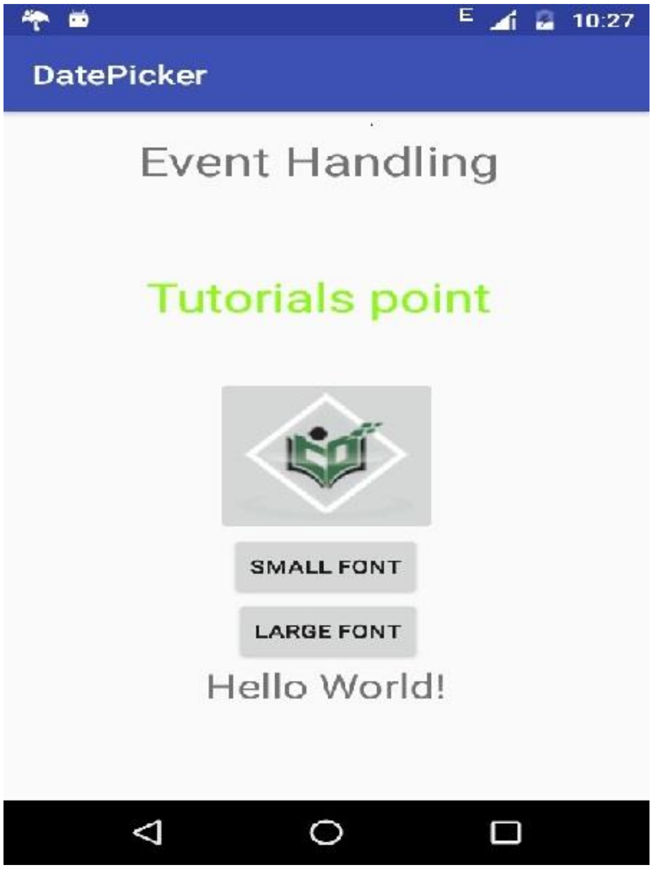
```
<?xml version="1.0" encoding="utf-8"?>
<resources>
<string name="app_name">myapplication</string>
</resources>
```

Following is the default content of AndroidManifest.xml −

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
package="com.example.myapplication" >

<application
android:allowBackup="true"
android:icon="@drawable/ic_launcher"
android:label="@string/app_name"
android:theme="@style/AppTheme" >

<activity
android:name="com.example.myapplication.MainActivity"
android:label="@string/app_name" >

<intent-filter>
<action android:name="android.intent.action.MAIN" />
<category android:name="android.intent.category.LAUNCHER" />
</intent-filter>

</activity>
```

```
</application>
</manifest>
```

OUTPUT:



click on two buttons, one by one and you will see that font of the **Hello World** text will change, which happens because registered click event handler method is being called against each click event.

RESULT :

Thus the Event Handling is implemented in android successfully.

EX.NO:14

LOCATION BASED SERVICES

AIM :

To write a program to obtain the current location by using location based services in android.

DESCRIPTON /ALGORITHM :

Android location APIs make it easy for you to build location-aware applications, without needing to focus on the details of the underlying location technology.

This becomes possible with the help of **Google Play services**, which facilitates adding location awareness to your app with automated location tracking, geofencing, and activity recognition

Step Description

1 You will use Android studio IDE to create an Android application and name it as Tutorialspoint under a package com.example.tutorialspoint7.myapplication.

2 add src/GPSTracker.java file and add required code.

3 Modify src/MainActivity.java file and add required code as shown below to take care of getting current location and its equivalent address.

4 Modify layout XML file res/layout/activity_main.xml to add all GUI components which include three buttons and two text views to show location/address.

5 Modify res/values/strings.xml to define required constant values

6 Modify AndroidManifest.xml as shown below

7 Run the application to launch Android emulator and verify the result of the changes done in the application.

PROGRAM :

Following is the content of the modified main activity file **MainActivity.java**.

```
package com.example.tutorialspoint7.myapplication;
```

```
importandroid.Manifest;
importandroid.app.Activity;
importandroid.os.Bundle;
import android.support.v4.app.ActivityCompat;
importandroid.test.mock.MockPackageManager;
importandroid.view.View;
importandroid.widget.Button;
importandroid.widget.Toast;

public class MainActivity extends Activity {

   Button btnShowLocation;
private static final int REQUEST_CODE_PERMISSION = 2;
   String mPermission = Manifest.permission.ACCESS_FINE_LOCATION;

   // GPSTracker class
GPSTrackergps;

   @Override
public void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);

try {
if (ActivityCompat.checkSelfPermission(this, mPermission)
            != MockPackageManager.PERMISSION_GRANTED) {

ActivityCompat.requestPermissions(this, new String[]{mPermission},
               REQUEST_CODE_PERMISSION);

            // If any permission above not allowed by user, this condition will
execute every time, else your else part will work
         }
      } catch (Exception e) {
e.printStackTrace();
      }

btnShowLocation = (Button) findViewById(R.id.button);

      // show location button click event
btnShowLocation.setOnClickListener(new View.OnClickListener() {
      @Override
public void onClick(View arg0) {
        // create class object
gps = new GPSTracker(MainActivity.this);

        // check if GPS enabled
if(gps.canGetLocation()){

double latitude = gps.getLatitude();
double longitude = gps.getLongitude();

          // \n is for new line
Toast.makeText(getApplicationContext(), "Your Location is - \nLat: "
            + latitude + "\nLong: " + longitude, Toast.LENGTH_LONG).show();
}else{
          // can't get location
          // GPS or Network is not enabled
          // Ask user to enable GPS/network in settings
gps.showSettingsAlert();
        }

      }
    });
  }
}
```

Following is the content of the modified main activity file **GPSTracker.java**.

```
package com.example.tutorialspoint7.myapplication;

importandroid.app.AlertDialog;
importandroid.app.Service;
importandroid.content.Context;
importandroid.content.DialogInterface;
importandroid.content.Intent;
importandroid.location.Location;
importandroid.location.LocationListener;
importandroid.location.LocationManager;
importandroid.os.Bundle;
importandroid.os.IBinder;
importandroid.provider.Settings;
importandroid.util.Log;

public class GPSTracker extends Service implements LocationListener {

private final Context mContext;

   // flag for GPS status
booleanisGPSEnabled = false;

   // flag for network status
booleanisNetworkEnabled = false;

   // flag for GPS status
booleancanGetLocation = false;

   Location location; // location
double latitude; // latitude
double longitude; // longitude

   // The minimum distance to change Updates in meters
private static final long MIN_DISTANCE_CHANGE_FOR_UPDATES = 10; //
10 meters

   // The minimum time between updates in milliseconds
private static final long MIN_TIME_BW_UPDATES = 1000 * 60 * 1; // 1 minute

   // Declaring a Location Manager
protectedLocationManagerlocationManager;

publicGPSTracker(Context context) {
this.mContext = context;
getLocation();
   }

public Location getLocation() {
try {
locationManager = (LocationManager)
mContext.getSystemService(LOCATION_SERVICE);
```

```
        // getting GPS status
isGPSEnabled =
locationManager.isProviderEnabled(LocationManager.GPS_PROVIDER);

        // getting network status
isNetworkEnabled = locationManager
          .isProviderEnabled(LocationManager.NETWORK_PROVIDER);

if (!isGPSEnabled&& !isNetworkEnabled) {
          // no network provider is enabled
        } else {
this.canGetLocation = true;
          // First get location from Network Provider
if (isNetworkEnabled) {
locationManager.requestLocationUpdates(
LocationManager.NETWORK_PROVIDER,
                MIN_TIME_BW_UPDATES,
                MIN_DISTANCE_CHANGE_FOR_UPDATES, this);

Log.d("Network", "Network");
if (locationManager != null) {
location = locationManager

.getLastKnownLocation(LocationManager.NETWORK_PROVIDER);

if (location != null) {
latitude = location.getLatitude();
longitude = location.getLongitude();
                }
             }
          }

          // if GPS Enabled get lat/long using GPS Services
if (isGPSEnabled) {
if (location == null) {
locationManager.requestLocationUpdates(
LocationManager.GPS_PROVIDER,
                  MIN_TIME_BW_UPDATES,
                  MIN_DISTANCE_CHANGE_FOR_UPDATES, this);

Log.d("GPS Enabled", "GPS Enabled");
if (locationManager != null) {
location = locationManager
                .getLastKnownLocation(LocationManager.GPS_PROVIDER);

if (location != null) {
latitude = location.getLatitude();
longitude = location.getLongitude();
              }
            }
          }
        }
      }

    } catch (Exception e) {
e.printStackTrace();
    }

return location;
  }

  /**
    * Stop using GPS listener
    * Calling this function will stop using GPS in your app
  * */

public void stopUsingGPS(){
if(locationManager != null){
locationManager.removeUpdates(GPSTracker.this);
    }
  }

  /**
    * Function to get latitude
  * */

public double getLatitude(){
if(location != null){
latitude = location.getLatitude();
    }

    // return latitude
```

```
return latitude;
   }

   /**
      * Function to get longitude
   * */

public double getLongitude(){
if(location != null){
longitude = location.getLongitude();
      }

      // return longitude
return longitude;
   }

   /**
      * Function to check GPS/wifi enabled
      * @return boolean
   * */

publicbooleancanGetLocation() {
returnthis.canGetLocation;
   }

   /**
      * Function to show settings alert dialog
      * On pressing Settings button will lauch Settings Options
   * */

public void showSettingsAlert(){
AlertDialog.BuilderalertDialog = new AlertDialog.Builder(mContext);

      // Setting Dialog Title
alertDialog.setTitle("GPS is settings");

      // Setting Dialog Message
alertDialog.setMessage("GPS is not enabled. Do you want to go to settings
menu?");

      // On pressing Settings button
```

```
alertDialog.setPositiveButton("Settings", new DialogInterface.OnClickListener()
{
public void onClick(DialogInterfacedialog,int which) {
            Intent intent = new
Intent(Settings.ACTION_LOCATION_SOURCE_SETTINGS);
mContext.startActivity(intent);
         }
      });

      // on pressing cancel button
alertDialog.setNegativeButton("Cancel", new DialogInterface.OnClickListener()
{
public void onClick(DialogInterface dialog, int which) {
dialog.cancel();
         }
      });

      // Showing Alert Message
alertDialog.show();
   }

   @Override
public void onLocationChanged(Location location) {
   }

   @Override
public void onProviderDisabled(String provider) {
   }

   @Override
public void onProviderEnabled(String provider) {
   }

   @Override
public void onStatusChanged(String provider, int status, Bundle extras) {
   }

   @Override
publicIBinderonBind(Intent arg0) {
return null;
   }
```

```
}
```

Following will be the content of **res/layout/activity_main.xml file** −

```
<?xml version = "1.0" encoding = "utf-8"?>
<LinearLayoutxmlns:android = "http://schemas.android.com/apk/res/android"
android:layout_width = "fill_parent"
android:layout_height = "fill_parent"
android:orientation = "vertical" >


<Button
android:id = "@+id/button"
android:layout_width = "fill_parent"
android:layout_height = "wrap_content"
android:text = "getlocation"/>

</LinearLayout>
```

Following will be the content of **res/values/strings.xml** to define two new constants −

```
<?xml version = "1.0" encoding = "utf-8"?>
<resources>
<string name = "app_name">Tutorialspoint</string>
</resources>
```

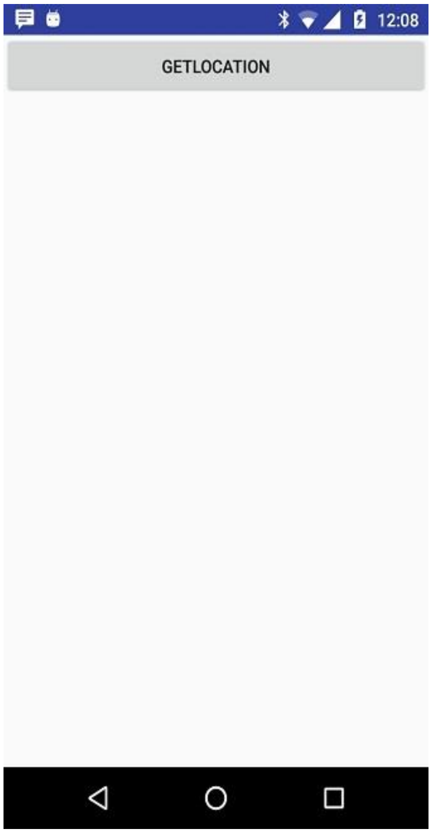
Following is the default content of **AndroidManifest.xml** −

```
<?xml version = "1.0" encoding = "utf-8"?>
<manifest xmlns:android = "http://schemas.android.com/apk/res/android"
package = "com.example.tutorialspoint7.myapplication">
<uses-permission android:name =
"android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name = "android.permission.INTERNET" />
<application
android:allowBackup = "true"
android:icon = "@mipmap/ic_launcher"
android:label = "@string/app_name"
android:supportsRtl = "true"
android:theme = "@style/AppTheme">
```

```
<activity android:name = ".MainActivity">
<intent-filter>
<action android:name = "android.intent.action.MAIN" />

<category android:name = "android.intent.category.LAUNCHER" />
</intent-filter>
</activity>
</application>
</manifest>
```

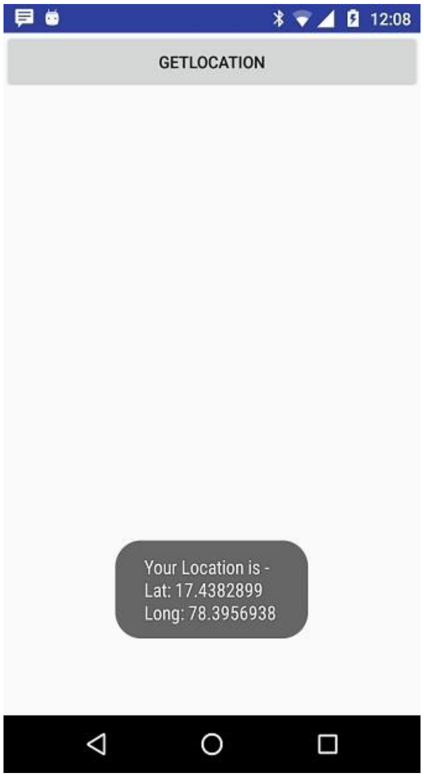
OUTPUT:

OUTPUT 1



Now to see location select **Get Location Button** which will display location information as follows −

OUTPUT 2



Your Location is -
Lat: 17.4382899
Long: 78.3956938

RESULT :

Thus the current location is obtained by using location based services in android successfully.

EX.NO:15

ACCESSING HARDWARE DEVICE – BLUETOOTH

AIM:

To Write a program to access the hardware device (Bluetooth) in android.

DESCRIPTON /ALGORITHM :

Among many ways, Bluetooth is a way to send or receive data between two different devices. Android platform includes support for the Bluetooth framework that allows a device to wirelessly exchange data with other Bluetooth devices.

Android provides Bluetooth API to perform these different operations.

- Scan for other Bluetooth devices
- Get a list of paired devices
- Connect to other devices through service discovery

Android provides BluetoothAdapter class to communicate with Bluetooth. Create an object of this calling by calling the static method getDefaultAdapter(). Its syntax is given below.

```
privateBluetoothAdapter BA;
BA = BluetoothAdapter.getDefaultAdapter();
```

In order to enable the Bluetooth of your device, call the intent with the following Bluetooth constant ACTION_REQUEST_ENABLE. Its syntax is.

```
Intent turnOn = new
Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
startActivityForResult(turnOn, 0);
```

Step	**Description**
1	You will use Android studio to create an Android application a package com.example.sairamkrishna.myapplication.
2	Modify src/MainActivity.java file to add the code
3	Modify layout XML file res/layout/activity_main.xml add any GUI

component if required.

4 Modify AndroidManifest.xml to add necessary permissions.

5 Run the application and choose a running android device and install the application on it and verify the results.

PROGRAM:

Here is the content of src/MainActivity.java

```
packagecom.example.sairamkrishna.myapplication;

importandroid.app.Activity;
importandroid.bluetooth.BluetoothAdapter;
importandroid.bluetooth.BluetoothDevice;

importandroid.content.Intent;
importandroid.os.Bundle;
importandroid.view.View;

importandroid.widget.ArrayAdapter;
importandroid.widget.Button;
importandroid.widget.ListView;

importandroid.widget.Toast;
importjava.util.ArrayList;
importjava.util.Set;

public class MainActivity extends Activity  {
  Button b1,b2,b3,b4;
privateBluetoothAdapter BA;
private Set<BluetoothDevice>pairedDevices;
ListView lv;

  @Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
```

```
      b1 = (Button) findViewById(R.id.button);
      b2=(Button)findViewById(R.id.button2);
      b3=(Button)findViewById(R.id.button3);
      b4=(Button)findViewById(R.id.button4);

      BA = BluetoothAdapter.getDefaultAdapter();
lv = (ListView)findViewById(R.id.listView);
   }

public void on(View v){
if (!BA.isEnabled()) {
        Intent turnOn = new
Intent(BluetoothAdapter.ACTION_REQUEST_ENABLE);
startActivityForResult(turnOn, 0);
Toast.makeText(getApplicationContext(), "Turned
on",Toast.LENGTH_LONG).show();
      } else {
Toast.makeText(getApplicationContext(), "Already on",
Toast.LENGTH_LONG).show();
      }
   }

public void off(View v){
BA.disable();
Toast.makeText(getApplicationContext(), "Turned off"
,Toast.LENGTH_LONG).show();
   }


public  void visible(View v){
      Intent getVisible = new
Intent(BluetoothAdapter.ACTION_REQUEST_DISCOVERABLE);
startActivityForResult(getVisible, 0);
   }
```

```
public void list(View v){
pairedDevices = BA.getBondedDevices();

ArrayList list = new ArrayList();

for(BluetoothDevicebt : pairedDevices) list.add(bt.getName());
Toast.makeText(getApplicationContext(), "Showing Paired
Devices",Toast.LENGTH_SHORT).show();

finalArrayAdapter adapter = new
ArrayAdapter(this,android.R.layout.simple_list_item_1, list);

lv.setAdapter(adapter);
   }
}
```

Here is the content of activity_main.xml

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
xmlns:tools="http://schemas.android.com/tools"
android:layout_width="match_parent"
android:layout_height="match_parent"
android:paddingLeft="@dimen/activity_horizontal_margin"
android:paddingRight="@dimen/activity_horizontal_margin"
android:paddingTop="@dimen/activity_vertical_margin"
android:paddingBottom="@dimen/activity_vertical_margin"
tools:context=".MainActivity"
android:transitionGroup="true">

<TextViewandroid:text="Bluetooth Example"
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:id="@+id/textview"
```

```
android:textSize="35dp"
android:layout_alignParentTop="true"
android:layout_centerHorizontal="true" />

<TextView
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Tutorials point"
android:id="@+id/textView"
android:layout_below="@+id/textview"
android:layout_centerHorizontal="true"
android:textColor="#ff7aff24"
android:textSize="35dp" />

<ImageView
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:id="@+id/imageView"
android:src="@drawable/abc"
android:layout_below="@+id/textView"
android:layout_centerHorizontal="true"
android:theme="@style/Base.TextAppearance.AppCompat" />

<Button
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Turn On"
android:id="@+id/button"
android:layout_below="@+id/imageView"
android:layout_toStartOf="@+id/imageView"
android:layout_toLeftOf="@+id/imageView"
android:clickable="true"
android:onClick="on" />

<Button
android:layout_width="wrap_content"
```

```
android:layout_height="wrap_content"
android:text="Get visible"
android:onClick="visible"
android:id="@+id/button2"
android:layout_alignBottom="@+id/button"
android:layout_centerHorizontal="true" />

<Button
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="List devices"
android:onClick="list"
android:id="@+id/button3"
android:layout_below="@+id/imageView"
android:layout_toRightOf="@+id/imageView"
android:layout_toEndOf="@+id/imageView" />

<Button
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="turn off"
android:onClick="off"
android:id="@+id/button4"
android:layout_below="@+id/button"
android:layout_alignParentLeft="true"
android:layout_alignParentStart="true" />

<ListView
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:id="@+id/listView"
android:layout_alignParentBottom="true"
android:layout_alignLeft="@+id/button"
android:layout_alignStart="@+id/button"
android:layout_below="@+id/textView2" />
```

```
<TextView
android:layout_width="wrap_content"
android:layout_height="wrap_content"
android:text="Paired devices:"
android:id="@+id/textView2"
android:textColor="#ff34ff06"
android:textSize="25dp"
android:layout_below="@+id/button4"
android:layout_alignLeft="@+id/listView"
android:layout_alignStart="@+id/listView" />

</RelativeLayout>
```

Here is the content of Strings.xml

```
<resources>
<string name="app_name">My Application</string>
</resources>
```

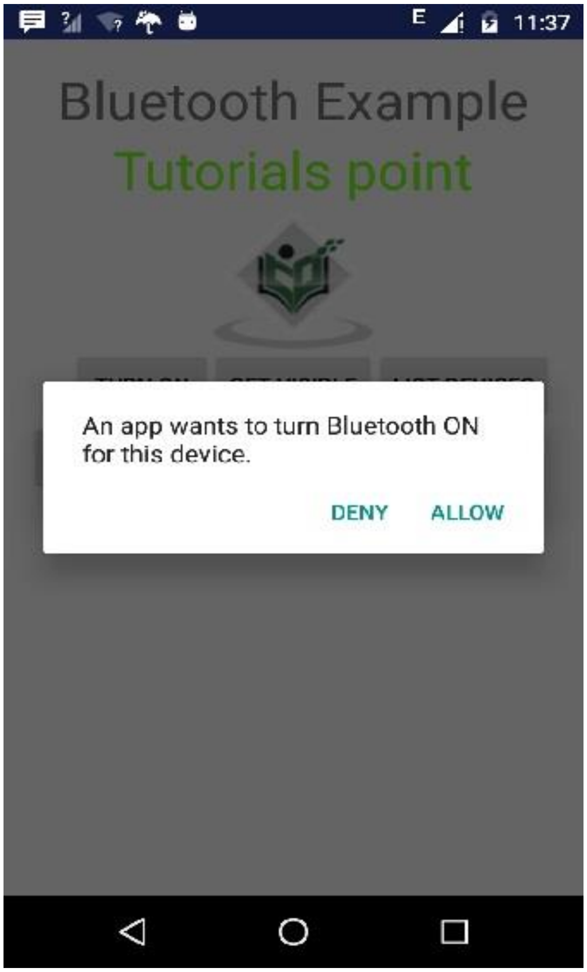
Here is the content of AndroidManifest.xml

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
package="com.example.sairamkrishna.myapplication" >
<uses-permissionandroid:name="android.permission.BLUETOOTH"/>
<uses-permissionandroid:name="android.permission.BLUETOOTH_ADMIN"/>

<application
android:allowBackup="true"
android:icon="@mipmap/ic_launcher"
android:label="@string/app_name"
android:theme="@style/AppTheme" >

<activity
android:name=".MainActivity"
android:label="@string/app_name" >
```

```
<intent-filter>
<action android:name="android.intent.action.MAIN" />
<category android:name="android.intent.category.LAUNCHER" />
</intent-filter>

</activity>

</application>
</manifest>
```

OUTPUT:

OUTPUT1



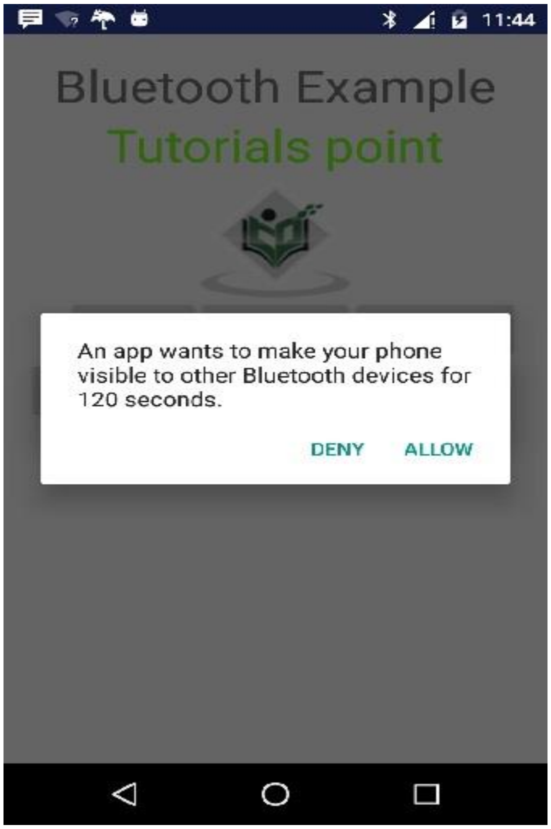
Bluetooth Example

Tutorials point

An app wants to turn Bluetooth ON for this device.

Now just select the Get Visible button to turn on your visibility. The following screen would appear asking your permission to turn on discovery for 120 seconds.

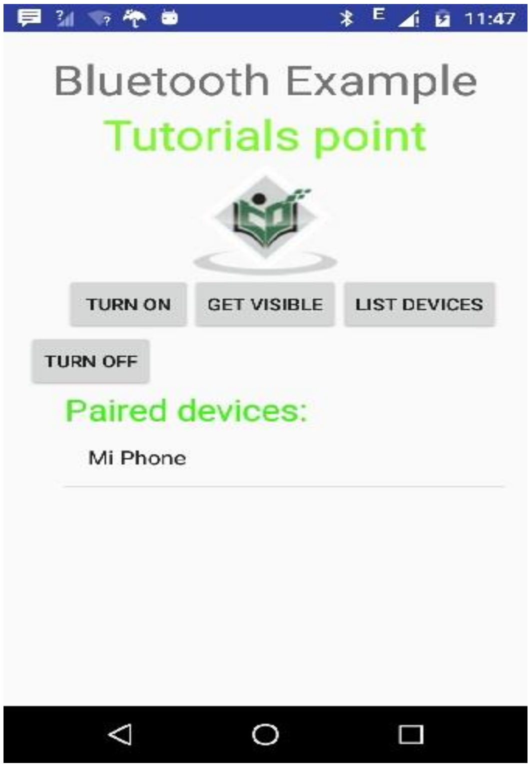
OUTPUT 2



An app wants to make your phone visible to other Bluetooth devices for 120 seconds.

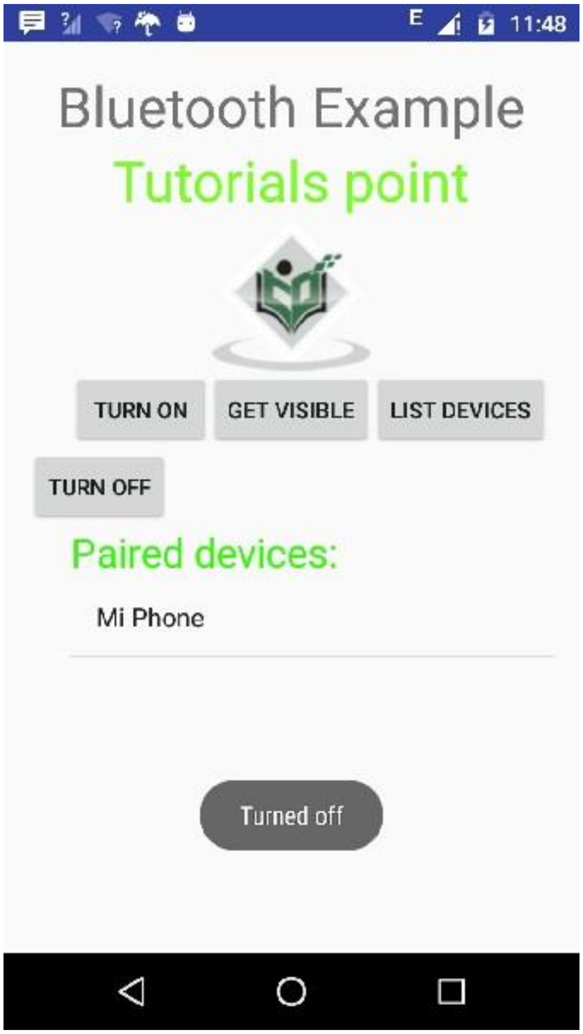
Now just select the List Devices option. It will list down the paired devices in the list view. In my case , I have only one paired device. It is shown below.

OUTPUT 3



Now just select the Turn off button to switch off the Bluetooth. Following message would appear when you switch off the bluetooth indicating the successful switching off of Bluetooth.

OUTPUT 4



Paired devices:

Mi Phone

Turned off

RESULT :

Thus the Bluetooth Hardware device is accessed in android successfully.

EX.NO:16

.FREQUENCY REUSE USING NS2

AIM:

To Develop scenario with mobile nodes to illustrate frequency reuse using ns2

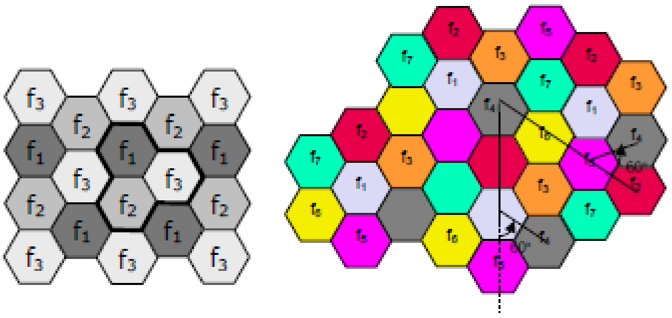
DESCRIPTON /ALGORITHM :

Cellular system for mobile communication implements SDM. Each transmitter typically called a base station covers a certain area called a cell.

However to avoid interference between transmitters which are within each others interference range, we use FDM. The general goal is never use the same frequency at the same time within the interference range. In order to do this , cells are grouped in to disjoint frequency units called clusters. The number of different frequencies used in a cluster is termed as frequency reuse factor N.

The following diagram shows two clusters one with 3 cells and one with 7 cells. We select frequency reuse factor N such a way that , the following condition satisfied.

$N = i^2 + i*j + j^2$



For N =3 =>i=1, j=1; For N=7 = >i=2 , j =1

Step Description

1. start
2. Accept value of number of cells per cluster i.e. N from the user.

3. Create a function that draws a hexagon.
4. Draw a clusters with N hexagons in each cluster.
5. Write appropriate frequencies in each cluster.
6. stop.

PROGRAM:

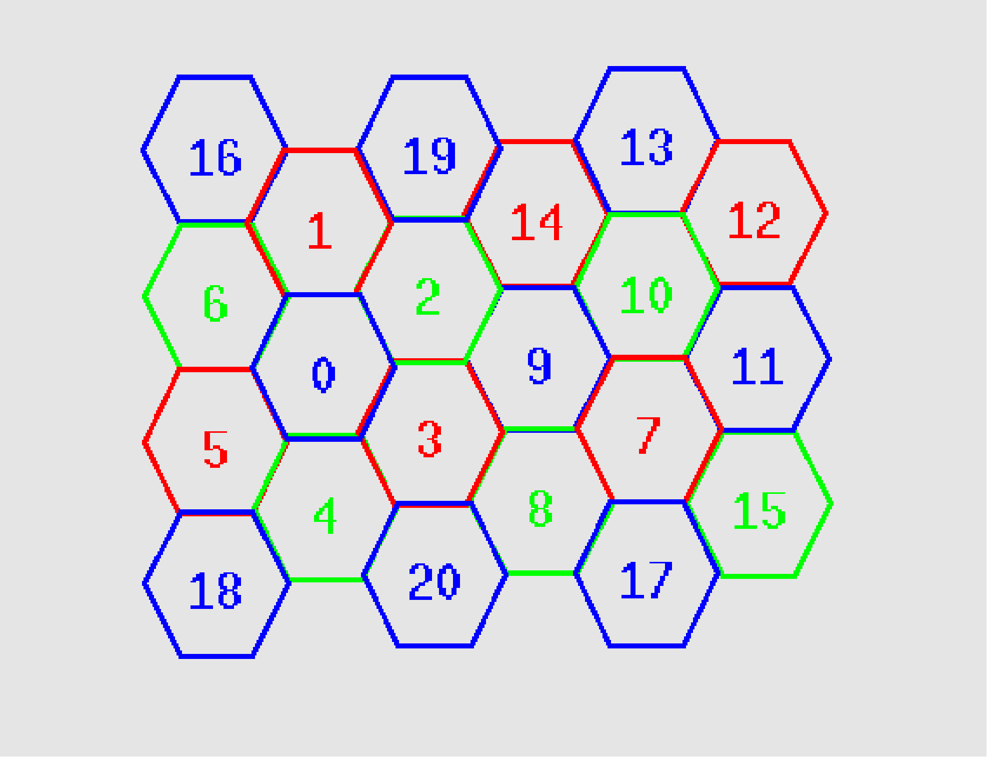
```
setval(chan) Channel/WirelessChannel ;# channel type
setval(prop) Propagation/TwoRayGround ;# radio-propagation model
setval(netif) Phy/WirelessPhy ;# network interface type
setval(mac) Mac/802_11 ;# MAC type
setval(ifq) Queue/DropTail/PriQueue ;# interface queue type
setval(ll) LL ;# link layer type
setval(ant) Antenna/OmniAntenna ;# antenna model
setval(ifqlen) 50 ;# max packet in ifq
setval(nn) 21 ;# number of mobilenodes
setval(rp) AODV ;# routing protocol
setval(x) 1052 ;# X dimension of topography
setval(y) 600 ;# Y dimension of topography
setval(stop) 10.0 ;# time of simulation end

#Create a ns simulator
set ns [new Simulator]

#Setup topography object
set topo [new Topography]
$topo load_flatgrid $val(x) $val(y)
create-god $val(nn)

#Open the NS trace file
settracefile [open out.tr w]
$ns trace-all $tracefile

#Open the NAM trace fi
setnamfile [open out.nam w]
$ns namtrace-all $namfile
$ns namtrace-all-wireless $namfile $val(x) $val(y)
setchan [new $val(chan)];#Create wireless channel


setval(0) 16
setval(1) 6
setval(2) 5
setval(3) 18
setval(4) 1
setval(5) 0
setval(6) 4
setval(7) 19
setval(8) 2
setval(9) 3
setval(10) 20
setval(11) 14
setval(12) 9
setval(13) 8
setval(14) 13
setval(15) 10
setval(16) 7
setval(17) 17
setval(18) 12
setval(19) 11
setval(20) 15

setclr(0) blue
setclr(1) green
setclr(2) red


for {set i 0 } { $i< 21 } {incri} {
        set n($i) [$ns node]
        $n($i) shape hexagon
        set index [expr $i % 3]


}

for {set i 0 } { $i< 21 } {incri} {

        setchk [expr $i % 7]
        if { $chk> 3 } {
        set index [expr [ expr $chk + 1 ] % 3]
        $n($val($i)) color $clr($index)
        } else {
```

```
        set index [expr $chk % 3]
        $n($val($i)) color $clr($index)
        }

}

$ns node-config -adhocRouting $val(rp) \
-llType $val(ll) \
-macType $val(mac) \
-ifqType $val(ifq) \
-ifqLen $val(ifqlen) \
-antType $val(ant) \
-propType $val(prop) \
-phyType $val(netif) \
-channel $chan \
-topoInstance $topo \
-agentTrace ON \
-routerTrace ON \
-macTrace ON \
-movementTrace ON \

$n(0) set X_ 303
$n(0) set Y_ 302
$n(0) set Z_ 0.0

$n(1) set X_ 302.74
$n(1) set Y_ 312.049
$n(1) set Z_ 0.0

$n(2) set X_ 310.333
$n(2) set Y_ 307.367
$n(2) set Z_ 0.0

$n(3) set X_ 310.428
$n(3) set Y_ 297.45
$n(3) set Z_ 0.0

$n(4) set X_ 303.12
$n(4) set Y_ 292.191
$n(4) set Z_ 0.0
```

```
$n(5) set X_ 295.501
$n(5) set Y_ 296.798
$n(5) set Z_ 0.0

$n(6) set X_ 295.552
$n(6) set Y_ 306.891
$n(6) set Z_ 0.0

$n(7) set X_ 325.684
$n(7) set Y_ 297.679
$n(7) set Z_ 0.0

$n(8) set X_ 318.174
$n(8) set Y_ 292.663
$n(8) set Z_ 0.0

$n(9) set X_ 317.998
$n(9) set Y_ 302.55
$n(9) set Z_ 0.0

$n(10) set X_ 325.547
$n(10) set Y_ 307.55
$n(10) set Z_ 0.0

$n(11) set X_ 333.246
$n(11) set Y_ 302.559
$n(11) set Z_ 0.0

$n(12) set X_ 333.01
$n(12) set Y_ 312.729
$n(12) set Z_ 0.0

$n(13) set X_ 325.524
$n(13) set Y_ 317.784
$n(13) set Z_ 0.0

$n(14) set X_ 317.877
$n(14) set Y_ 312.654
$n(14) set Z_ 0.0

$n(15) set X_ 333.35
```

```
$n(15) set Y_ 292.476
$n(15) set Z_ 0.0

$n(16) set X_ 295.477
$n(16) set Y_ 317.135
$n(16) set Z_ 0.0

$n(17) set X_ 325.523
$n(17) set Y_ 287.639
$n(17) set Z_ 0.0

$n(18) set X_ 295.55
$n(18) set Y_ 286.907
$n(18) set Z_ 0.0

$n(19) set X_ 310.411
$n(19) set Y_ 317.204
$n(19) set Z_ 0.0


$n(20) set X_ 310.744
$n(20) set Y_ 287.567
$n(20) set Z_ 0.0

proc finish {} {
global ns tracefilenamfile
$ns flush-trace
close $tracefile
close $namfile
execnamout.nam&
exit 0
}
for {set i 0} {$i< $val(nn) } { incri } {
$ns at $val(stop) "\$n($i) reset"
}
$ns at $val(stop) "$ns nam-end-wireless $val(stop)"
$ns at $val(stop) "finish"
$ns at $val(stop) "puts \"done\" ; $ns halt"
$ns run
```

OUTPUT:



RESULT:

The frequency reuse is implemented in NS2 successfully.