Expno:1 Write a java program to implement the concept of

a)authentication:

```
import java.io.*;

import java.util.*;

public class authen

{

public static void main(String args[]){

Scanner sc=new Scanner(System.in);

String user1="arun";

String pass1="1234";

System.out.println("username");

String username=sc.nextLine();

System.out.println("password");

String password=sc.nextLine();

if(username.equals(user1) && password.equals(pass1))

{

    System.out.println("validuser");

}

else{

 System.out.println("pleas enter a valid credentials");
}

}

}
```

b)autherization.

```
import java.io.*;

import java.util.*;

public class exp1b {

private static String nextLine;

public static void main(String args[]) throws IOException{

Scanner sc = new Scanner(System.in);

        String user1 = "arun";

        String pass1 = "arun";

        String admin = "pradeesh";

        String apass = "pradeesh";

        System.out.println("username");

        String uname = sc.nextLine();

        System.out.println("password");

        String pass = sc.nextLine();


        if(uname.equals(user1)&&pass.equals(pass1)){
```

```
    System.out.println("Authentication Successfull, welcome user");

    File myobj = new File("authfile.txt");

    Scanner myReader= new Scanner(myobj);

    while(myReader.hasNextLine()){

        String data = myReader.nextLine();

        System.out.println(data);

    }

    myReader.close();

}

else if(uname.equals(admin)&&pass.equals(apass)){

    System.out.println("Welcome admin");

    System.out.println("enter 1 for read and 2 for write");

    int choice = sc.nextInt();

    switch(choice){

        case 1:

            File myobj = new File("authfile.txt");

            Scanner myReader= new Scanner(myobj);

            while(myReader.hasNextLine()){

            nextLine = myReader.nextLine();

            String data = nextLine;
```

```
                    System.out.println(data);

                    }

                    myReader.close();

                    break;

                case 2:

                    FileWriter mywriter = new FileWriter("authfile.txt");

                    mywriter.write("i am your admin");

                    mywriter.close();

                    break;

            }

        }

        else {

            System.out.println("Enter valid credentials");

        }

    }

}
```

Expno:2 active and passive attack

Program:

```
import java.io.*;

import java.util.*;

public class active {

   public static void main(String args[]) throws IOException {

    Scanner input = new Scanner(System.in);

    System.out.println("enter the line or word");

    String content=input.nextLine();

    System.out.println("Enter 1 for accessing file ,2 for testing attacks");

    int x = input.nextInt();

    switch(x){

      case 1 -> {

        FileWriter mywriter=new FileWriter("file.txt");



        String data;

        mywriter.write(content);

        mywriter.close();

        System.out.println("Content sucessfully tranfered");

        //Reading file2
    File obj=new File("file.txt");

    Scanner myReader=new Scanner(obj);

    while(myReader.hasNextLine()){

        data=myReader.nextLine();

        System.out.println(data);

        if(content.equals(data)){

            System.out.println("Passive Attack");

        }

        else{

            System.out.println("Active Attack");

        }}

    myReader.close();

}


case 2 -> {

  File obj1=new File("file.txt");

  Scanner myReader1=new Scanner(obj1);

  while(myReader1.hasNextLine()){

  String data=myReader1.nextLine();

  System.out.println(data);
```

```
          if(content.equals(data)){
            System.out.println("Passive Attack");
          }
          else{
          System.out.println("Active Attack");
          }
          }
          myReader1.close();
         }
      }
}
   }
```

Expno:3 write a java program to implement Euclidian algorithm and modular arithmetic

1)modular arithmetic:

```
import java.io.*;
import java.util.*;
public class exp3{
   public static void main(String args[])
   {
```

```
   while(1>0){
    System.out.print("Enter The Arithmetic Modular Value");
    Scanner sc=new Scanner(System.in);
    int s=sc.nextInt();
    System.out.print(" 1.ADDITON \n 2.SUBRACTION \n 
3.MULTIPLICATION   \n Enter The Number for Above Options \n");
    int n=sc.nextInt();
switch(n)
{
  case 1:
     System.out.println("This is Additon operation modular value is "+s);
     for(int i=0;i<s;i++){
       for(int j=0;j<s;j++){
         int c=i+j;
           if(c>(s-1))
           {
            c=c%s;
            System.out.print("\t"+c);
           }
          else
```

```
            {
              System.out.print("\t"+c);
            }
          }
       System.out.print("\n");
    }
    break;


case 2:
System.out.println("This is subraction operation modular value is "+s);
    for(int i=0;i<s;i++){
       for(int j=0;j<s;j++){
          int c=i-j;
          if(c>=0){
            if(c>(s-1))
            {
              c=c%s;
              System.out.print("\t"+c);
            }
          else
          {
             System.out.print("\t"+c);
          }
        }
        else{
          c=c+s;
          if(c>(s-1))
          {
            c=c%s;
            System.out.print("\t"+c);
          }
        else
        {
           System.out.print("\t"+c);
        }
        }
          }

System.out.print("\n");
 }
```

```
    break;


  case 3:
       System.out.println("This is multipli operation modular value is "+s);
       for(int i=1;i<s;i++){
           for(int j=1;j<s;j++){
               int c=i*j;
                  if(c>(s-1))
                  {
                    c=c%s;
                    if(c==0){
                      c=1;
                      System.out.print("\t"+c);
                      }
                      else{
                    System.out.print("\t"+c);
                      }
                  }
                else
                {
                       System.out.print("\t"+c);
                    }
                  }
        System.out.print("\n");
        }
     break;
    default :
          System.out.println("THANK YOU FOR USING MY
CALCULATION");
}
}
}
}
```

b)Euclidian algorithm:

```
import java.io.*;
import java.util.*;
public class gcd {
    public static void main(String args[]){
        Scanner sc=new Scanner(System.in);
        while(true){
            System.out.println("\n");
System.out.println("Enter the  value of a");
int a=sc.nextInt();
System.out.println("Enter the value of b");
int b=sc.nextInt();
for(int i=1;i<1000;i++){
    if(a==0&&b==0){
        System.out.println("the gcd vlaue is 1");
        break;
    }
    if(a>b){
    int r=a%b;
    System.out.println("the step \t"+i);
    System.out.println("the Remainder value is \t"+r);
    if(r>0){
        a=b;
        b=r;
    }
    if(r==0){
    System.out.println("the a value is \t"+a);
    System.out.println("the b value is \t"+b);
```

```
            System.out.print("GCD Value is ("+a);

            System.out.print(",");

            System.out.print(+b);

            System.out.print(")");

            System.out.print("="+b);

              break;

           }

        }

        else{

           System.out.println("step"+i);

           System.out.println("swap is  processing");

            int c=a;

            a=b;

            b=c;

        }

     }

     }

     }

  }
```

Expno:4 encryption and decryption using ceaser cipher(substitution technique)

Program:

```
import java.io.*;
import java.util.*;
import java.awt.Point;
public class cc1{
    cc1(){
     char a[]={'A','B','C','D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U','V','W','X','Y','Z'};
       Scanner sc=new Scanner(System.in);
       System.out.println("CEASER CIPHER");
       System.out.println("Enter the pt :");
       String pt=sc.nextLine();
       System.out.println("Enter the key :");
       int key=sc.nextInt();
       char[] cs=new char[pt.length()];
       for(int i=0;i<pt.length();i++){
        cs[i]=pt.charAt(i);
       }
       for(char c:cs){
```

```
System.out.print("\t"+c);

}

System.out.print("\n");

for(int i=0;i<cs.length;i++){

 for(int j=0;j<a.length;j++){

     if(cs[i]==a[j]){

         int s=j+key;

      if(s<a.length){

         System.out.println(cs[i]+"+"+key+"="+a[s]);



      }

      else{

         int k=a.length;

          s=s-k;

          System.out.println(cs[i]+"+"+key+"="+a[s]);

      }

      }



     }
```

```
}

System.out.println("----------------------------------------------");

    System.out.println("Decrypt");

    System.out.println("----------------------------------------------");

    for(int i=0;i<cs.length;i++){

        for(int j=0;j<a.length;j++){

            if(cs[i]==a[j]){

                int s=j+key;

                int o=s-key;

             if(s<a.length){

                System.out.println(a[s]+"-"+key+"="+a[o]);

             }

             else{

                int k=a.length;

                s=s-k;

                System.out.println(a[s]+"+"+key+"="+a[o]);

             }

             }



            }
            }



        }

        private static char[][] charTable;

private static Point[] positions;

private static String prepareText(String s, boolean chgJtoI) {

s = s.toUpperCase().replaceAll("[^A-Z]", "");

return chgJtoI ? s.replace("J", "I") : s.replace("Q", "");

}

private static void createTbl(String key, boolean chgJtoI) {

charTable = new char[5][5];

positions = new Point[26];

String s = prepareText(key + "ABCDEFGHIJKLMNOPQRSTUVWXYZ",
chgJtoI);

StringBuilder sb1=new StringBuilder();

String s1="ABCDEFGHIKLMNOPQRSTUVWXYZ";

String s2=key.concat(s1);

s2.chars().distinct().forEach(c -> sb1.append((char)c));

char[] ch=new char[sb1.length()];

for(int i=0;i<sb1.length();i++){

 ch[i]=sb1.charAt(i);
```

```
}
int n=0;
char ci[][]=new char[5][5];
for(int i=0;i<5;i++){
    for(int j=0;j<5;j++){
        if(n<26)
        {
         ci[i][j]=ch[n];
         System.out.print("\t"+ci[i][j]);
         n++;
        }
    }
    System.out.print("\n");
}
int len = s.length();
for (int i = 0, k = 0; i < len; i++) {
char c = s.charAt(i);
if (positions[c - 'A'] == null) {
charTable[k / 5][k % 5] = c;
positions[c - 'A'] = new Point(k % 5, k / 5);
```

```
k++;

}

}

}

private static String codec(StringBuilder txt, int dir) {

int len = txt.length();

for (int i = 0; i < len; i += 2) {

char a = txt.charAt(i);

char b = txt.charAt(i + 1);

int row1 = positions[a - 'A'].y;

int row2 = positions[b - 'A'].y;

int col1 = positions[a - 'A'].x;

int col2 = positions[b - 'A'].x;

if (row1 == row2) {

col1 = (col1 + dir) % 5;

col2 = (col2 + dir) % 5;

} else if (col1 == col2) {

row1 = (row1 + dir) % 5;

row2 = (row2 + dir) % 5;

} else {
```

```
int tmp = col1;

col1 = col2;

col2 = tmp;

}

txt.setCharAt(i, charTable[row1][col1]);

txt.setCharAt(i + 1, charTable[row2][col2]);

}

return txt.toString();

}

private static String encode(String s) {

StringBuilder sb = new StringBuilder(s);

for (int i = 0; i < sb.length(); i += 2) {

if (i == sb.length() - 1) {

sb.append(sb.length() % 2 == 1 ? 'X' : "");

} else if (sb.charAt(i) == sb.charAt(i + 1)) {

sb.insert(i + 1, 'X');

}

}

return codec(sb, 1);

}
```

```
private static String decode(String s) {
return codec(new StringBuilder(s), 4);
}
public static int[][] keymat = new int[][] { { 1, 2, 1 }, { 2, 3, 2 },
        { 2, 2, 1 } };
        public static int[][] invkeymat = new int[][] { { -1, 0, 1 }, { 2, -1, 0 }, { -2, 2, -1 } };
        public static String key = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
        private static String encode(char a, char b, char c) {
        String ret = "";
        int x, y, z;
        int posa = (int) a - 65;
        int posb = (int) b - 65;
        int posc = (int) c - 65;
        x = posa * keymat[0][0] + posb * keymat[1][0] + posc * keymat[2][0];
        y = posa * keymat[0][1] + posb * keymat[1][1] + posc * keymat[2][1];
        z = posa * keymat[0][2] + posb * keymat[1][2] + posc * keymat[2][2];
        a = key.charAt(x % 26);
        b = key.charAt(y % 26);
        c = key.charAt(z % 26);
        ret = "" + a + b + c;
```

```
return ret;

}

private static String decode(char a, char b, char c) {

String ret = "";

int x, y, z;

int posa = (int) a - 65;

int posb = (int) b - 65;

int posc = (int) c - 65;

x = posa * invkeymat[0][0] + posb * invkeymat[1][0] + posc * invkeymat[2][0];

y = posa * invkeymat[0][1] + posb * invkeymat[1][1] + posc * invkeymat[2][1];

z = posa * invkeymat[0][2] + posb * invkeymat[1][2] + posc * invkeymat[2][2];

a = key.charAt((x % 26 < 0) ? (26 + x % 26) : (x % 26));

b = key.charAt((y % 26 < 0) ? (26 + y % 26) : (y % 26));

c = key.charAt((z % 26 < 0) ? (26 + z % 26) : (z % 26));

ret = "" + a + b + c;

return ret;

}

static String encode(String text, final String key) {
```

```
String res = "";
text = text.toUpperCase();
for (int i = 0, j = 0; i < text.length(); i++) {
char c = text.charAt(i);
if (c < 'A' || c > 'Z') {
continue;
}
res += (char) ((c + key.charAt(j) - 2 * 'A') % 26 + 'A');
j = ++j % key.length();
}
return res;
}
static String decode(String text, final String key) {
String res = "";
text = text.toUpperCase();
for (int i = 0, j = 0; i < text.length(); i++) {
char c = text.charAt(i);
if (c < 'A' || c > 'Z') {
continue;
}
```

```
    res += (char) ((c - key.charAt(j) + 26) % 26 + 'A');

    j = ++j % key.length();

    }

    return res;

    }

      public static void main(String args[]) throws java.lang.Exception {

     while (1>0){

         Scanner sc=new Scanner(System.in);

         System.out.println("enter the operation \n 1.ceaser cipher \n 2.playfair
cipher \n 3.hill cipher \n 4.vigenere cipher");

         int g=sc.nextInt();

         switch(g){

           case 1:

              cc1 c=new cc1();

              break;

  case 2:

  Scanner si=new Scanner(System.in);

    System.out.println("PLAYFAIR CIPHER");

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

    String key =si.nextLine();

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

```
    String txt = si.nextLine();
    boolean chgJtoI = true;
    createTbl(key, chgJtoI);
    String enc = encode(prepareText(txt, chgJtoI));
    System.out.println("Simulating Playfair Cipher\n----------------------");
    System.out.println("Input Message : " + txt);
    System.out.println("Encrypted Message : " + enc);
    System.out.println("Decrypted Message : " + decode(enc));
    break;
  case 3:
  Scanner sT=new Scanner(System.in);
    String msg;
    String ecc = "";
    String dec = "";
    int n;
    System.out.println("HILL CIPHER");
    System.out.println("Enter the input");
  msg =sT.nextLine();
System.out.println("simulation of Hill Cipher\n-------------------------");
System.out.println("Input message : " + msg);
```

```
msg = msg.toUpperCase();

msg = msg.replaceAll("\\s", "");

n = msg.length() % 3;

if (n != 0) {

for (int i = 1; i <= (3 - n); i++) {

msg += 'X';

}

}

System.out.println("padded message : " + msg);

char[] pdchars = msg.toCharArray();

for (int i = 0; i < msg.length(); i += 3) {

ecc += encode(pdchars[i], pdchars[i + 1], pdchars[i + 2]);

}

System.out.println("encoded message : " + ecc);

char[] dechars = ecc.toCharArray();

for (int i = 0; i < ecc.length(); i += 3) {

dec += decode(dechars[i], dechars[i + 1], dechars[i + 2]);

}

System.out.println("decoded message : " + dec);

break;
```

```
 case 4:
 System.out.println("VIGNERECIPHER");
 String kay = "VIGENERECIPHER";
 String msg1 = "SecurityLaboratory";
  System.out.println("Simulating Vigenere Cipher\n------------------------");
  System.out.println("Input Message : " + msg1);
  String encc = encode(msg1, kay);
  System.out.println("Encrypted Message : " + encc);
  System.out.println("Decrypted Message : " + decode(encc, kay));
  break;
 }
}
}
}
```

Expno:5 rail fence cipher ,row and column (transposition technique)

Program:

```
import java.io.*;
import java.util.*;
public class all5exp1 {
    all5exp1(){
        Scanner sc=new Scanner(System.in);
        String s1="siva is good boy";
        int m=3;
        s1=s1.replaceAll("\\s","");
        int k=0;
        char[] ch=new char[s1.length()];
for(int i=0;i<s1.length();i++){
 ch[i]=s1.charAt(i);
}
char ci[][]=new char[100][100];
for(int i=0;i<=10;i++){
    for(int j=1;j<=m;j++){
         if(k< ch.length){
            ci[i][j]=ch[k];
            System.out.print(ch[k]);
            k=k+1;
        }
        else{
            break;
        }

            }

    System.out.print("\t");

}

System.out.print("\n");

System.out.print("Encrypt:");

for(int j=1;j<=m;j++){

for(int i=0;i<ci.length;i++){

         System.out.print(ci[i][j]);

    }

}

System.out.print("\n");

System.out.print("Decrypt:");

for(int i=0;i<ci.length;i++){

for(int j=1;j<=m;j++){



            System.out.print(ci[i][j]);

        }



    }
    }
public static void main(String args[])throws java.lang.Exception{
    while(true){
        Scanner sc=new Scanner(System.in);
        System.out.println("\n Enter the operation \n 1.rail fence \n 2.row and
coloumn");
        int g=sc.nextInt();
            if(g==1){
                  all5exp1 s=new all5exp1();}
            else if(g==2){
                  second p=new second();
            }
            else{
                System.out.println("Enter valid crendtials");
            }
        }
    }
        }
class second{
    second(){
        String pl = "i am ramchandru";
          int q=5;
pl=pl.toUpperCase();
pl=pl.replaceAll("\\s","");
//matrix
int h=1;
String s1="ABCDEFGHIJKLMNOPQRSTUVWXYZ";
String s2=pl.concat(s1);
char[] cha=new char[s2.length()];
for(int i=0;i<s2.length();i++){
 cha[i]=s2.charAt(i);
}
String key="6521374";
char[] kay=new char[key.length()];
for(int i=0;i<key.length();i++){
    kay[i]=key.charAt(i);
}
for(char l:kay)
{
System.out.print("\t"+l);
```

```
}

System.out.print("\n");

int n=0;

char ci[][]=new char[20][20];

char so[][]=new char[20][20];

for(int i=0;i<q;i++){

    for(int j=0;j<kay.length;j++){

        if(n<s2.length())

        {

         ci[i][j]=cha[n];

         System.out.print("\t"+ci[i][j]);

         n++;

        }

    }

    System.out.print("\n");

}

System.out.print("\n");

System.out.print("\n");

for(int i=0;i<q;i++){

    for(int j=0;j<kay.length;j++){
        int a=Character.getNumericValue(kay[j]);
        if(a==h){
            for(int k=0;k<kay.length;k++){
               so[i][k]=ci[k][j];
               System.out.print("\t"+so[i][k]);

            }
            System.out.print("\n");
            h=h+1;
        }
    }
}
System.out.print("The cipher text value is:");
int o=1;
for(int i=0;i<q;i++){
    for(int j=0;j<kay.length;j++){
        int a=Character.getNumericValue(kay[j]);
        if(a==o){
            for(int k=0;k<kay.length;k++){
               so[i][k]=ci[k][j];
               System.out.print(so[i][k]);
            }
            o=o+1;
        }
    }
}
    }
}
```

Expno:6 steganography

Program:

```
// Java code for watermarking an image

// For setting color of the watermark text
import java.awt.Color;

// For setting font of the watermark text
import java.awt.Font;
import java.awt.Graphics;
import java.awt.image.BufferedImage;
import java.io.File;
import java.io.IOException;
import javax.imageio.ImageIO;

public class watermark {
    public static void main(String[] args)
    {
        BufferedImage img = null;
        File f = null;

        // Read image
        try {
            f = new File(
                "C:\\Users\\BRAJASANKAR
R\\Pictures\\DSC_0445.JPG");
            img = ImageIO.read(f);
     System.out.println("* Image is successfully readed");
        }
        catch (IOException e) {
            System.out.println(e);
        }
```

```
// create BufferedImage object of same width and
// height as of input image
BufferedImage temp = new BufferedImage(
        img.getWidth(), img.getHeight(),
        BufferedImage.TYPE_INT_RGB);

// Create graphics object and add original
// image to it
Graphics graphics = temp.getGraphics();
graphics.drawImage(img, 0, 0, null);

// Set font for the watermark text
graphics.setFont(new Font("Arial", Font.PLAIN, 80));
graphics.setColor(new Color(255, 0, 0, 40));

// Setting watermark text
String watermark = "WaterMark generated";

// Add the watermark text at (width/5, height/3)
```

```
                // location

                graphics.drawString(watermark, img.getWidth() / 5,

                                                        img.getHeight() / 3);


                // releases any system resources that it is using

                graphics.dispose();

      System.out.println("* Successfully watermark generated");

                f = new File("C:\\Users\\BRAJASANKAR R\\Pictures\\siva.JPG");

                try {

                        ImageIO.write(temp, "jpg", f);

          System.out.println("* the process is done  \n* go and check the given location");

                }

                catch (IOException e) {

                        System.out.println(e);

                }

        }

}
```

Expno:7 DES Algorithm

Program:

```
import java.security.InvalidKeyException;

import java.security.NoSuchAlgorithmException;

import javax.crypto.BadPaddingException;

import javax.crypto.Cipher;

import javax.crypto.IllegalBlockSizeException;

import javax.crypto.KeyGenerator;

import javax.crypto.NoSuchPaddingException;

import javax.crypto.SecretKey;

import java.util.*;

public class DES {

    public static void main(String[] argv) {

        try{

            Scanner v=new Scanner(System.in);

            System.out.println("Message Encryption Using DES Algorithm\n-------
");

            KeyGenerator keygenerator = KeyGenerator.getInstance("DES");

            SecretKey myDesKey = keygenerator.generateKey();

            Cipher desCipher;

            desCipher = Cipher.getInstance("DES/ECB/PKCS5Padding");

            desCipher.init(Cipher.ENCRYPT_MODE, myDesKey);
            //byte[] text = "Secret Information ".getBytes();

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

            String b=v.nextLine();

            byte[] text=b.getBytes();

            System.out.println("Message [Byte Format] : " + text);

            System.out.println("Message : " + new String(text));

            byte[] textEncrypted = desCipher.doFinal(text);

            System.out.println("Encrypted Message: " + textEncrypted);

            desCipher.init(Cipher.DECRYPT_MODE, myDesKey);

            byte[] textDecrypted = desCipher.doFinal(textEncrypted);

            System.out.println("Decrypted Message: " + new
String(textDecrypted));

            }

            catch(NoSuchAlgorithmException e){

                e.printStackTrace();

                }

            catch(NoSuchPaddingException e){

                e.printStackTrace();

                }

                catch(InvalidKeyException e){

                    e.printStackTrace();
```

```
          }
        catch(IllegalBlockSizeException e){
          e.printStackTrace();
        }
        catch(BadPaddingException e){
          e.printStackTrace();
          }
        }
    }
```

Expno:8 AES Algorithm

Program:

```
import java.io.UnsupportedEncodingException;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import java.util.Arrays;
import java.util.Base64;
import javax.crypto.Cipher;
import javax.crypto.spec.SecretKeySpec;
import java.util.*;
```

```
public class AES {

private static SecretKeySpec secretKey;

private static byte[] key;

public static void setKey(String myKey)

{

   MessageDigest sha = null;

   try {

   key = myKey.getBytes("UTF-8");

   sha = MessageDigest.getInstance("SHA-1");

   key = sha.digest(key);

   key = Arrays.copyOf(key, 16);

   secretKey = new SecretKeySpec(key, "AES");

   }

   catch (NoSuchAlgorithmException e) {

       e.printStackTrace();

       }

   catch (UnsupportedEncodingException e) {

      e.printStackTrace();

      }

 }
 public static String encrypt(String strToEncrypt, String secret) {
   try {
      setKey(secret);
      Cipher cipher = Cipher.getInstance("AES/ECB/PKCS5Padding");
      cipher.init(Cipher.ENCRYPT_MODE, secretKey);
      return

Base64.getEncoder().encodeToString(cipher.doFinal(strToEncrypt.getBytes("UTF8")));
      }
      catch (Exception e) {
          System.out.println("Error while encrypting: " + e.toString());
          }
          return null;
      }
      public static String decrypt(String strToDecrypt, String secret) {
          try {
              setKey(secret);
              Cipher cipher = Cipher.getInstance("AES/ECB/PKCS5PADDING");
              cipher.init(Cipher.DECRYPT_MODE, secretKey);
        return new
String(cipher.doFinal(Base64.getDecoder().decode(strToDecrypt)));

        }

        catch (Exception e) {

          System.out.println("Error while decrypting: " + e.toString());

          }

          return null;

          }

          public static void main(String[] args) {

            //final String secretKey = "ganeshuniverssit";

           // String originalString = "www.indgan.com";

            Scanner s=new Scanner(System.in);

            System.out.println("enter the secret key");

            final String secretKey=s.nextLine();

            System.out.println("enter the final string");

            String originalString=s.nextLine();

            String encryptedString = AES.encrypt(originalString,
secretKey);

            String decryptedString = AES.decrypt(encryptedString,
secretKey);

            System.out.println("Encryption Using AES Algorithm\n-----------
-");
```

```
                        System.out.println("Original URL : " + originalString);

                        System.out.println("Encrypted URL : " + encryptedString);

                        System.out.println("Decrypted URL : " + decryptedString);

                    }

}
```

Expno:9 Electroic code book(ECB)

Program:

```
import java.util.*;

class ECB{

  public static void main(String[] args){

     Scanner sc=new Scanner(System.in);

     Scanner so=new Scanner(System.in);

     String p[][]=new String[100][100];

     System.out.println("how many text is giving :");

     int h=sc.nextInt();

     int i;

     int q=1;

     int e=0;

     for(i=0;i<h;i++){
      System.out.println("Enter the string"+(i+1)+"=:");
      p[q][i]=sc.next();
      q++;
     }
     System.out.println("enter the key value:");
     String key=so.nextLine();
     System.out.println(key);
     System.out.println();
     char c[][]=new char[100][100];
     char[]
alpha1={'A','B','C','D','E','F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U','V','
W','X','Y','Z'};
     char[] alpha=new char[100];
     for(i=0;i<26;i++){
      alpha[i+1]=alpha1[i];
     }
     char check[]=new char[100];
     char ks[]=new char[100];
    /*encryption */
    q=1;
    for(int j=0;j<h;j++)
```

```
{
 p[q][j]=p[q][j].toUpperCase();
  for(i=0;i<p[q][j].length();i++)
  {
    check[i]=p[q][j].charAt(i);
      for(int k=1;k<=26;k++)
      {
        if(alpha[k]==check[i])
         {
            key=key.toUpperCase();
            for(int l=0;l<key.length();l++)
            {
             ks[l]=key.charAt(l);
             for(int w=1;w<=26;w++)
              {

              if(alpha[w]==ks[l])
               {
                 int o=k+w;
                 if(o<=26){
```

```
               c[q][i]=alpha[o];


               }
               else{
                 o=o%26;
                 c[q][i]=alpha[o];




               }
               break;
             }
            }
            break;
           }
       break;}
     }
  }
  q++;
}
System.out.println("encrypted values:");
```

```
for(int j=1;j<=h;j++){
  for(i=0;i<p.length;i++){
    System.out.print(c[j][i]);
  }
  System.out.println("");
}  /*encryption done */
/*decryption start */
System.out.print("\n");

for(q=1;q<=h;q++){
 for(i=0;i<c.length;i++)
 {
    for(int k=1;k<=26;k++)
     {
       if(alpha[k]==c[q][i])
        {
           key=key.toUpperCase();
           for(int l=0;l<key.length();l++)
           {
```

```
ks[l]=key.charAt(l);

for(int w=1;w<=26;w++)

 {

 if(alpha[w]==ks[l])

  {

    k=k+26;

    if(k<=26){

    c[q][i]=alpha[k];

         }

    else{

    int o=k-w;

    if(o<=26){

     c[q][i]=alpha[o];


    }

    else{

      o=o%26;

      c[q][i]=alpha[o];

                                        }
```

```
                   }
                   break;
                 }
                }
                break;
               }
             }
          }
        }
      }
System.out.println("Decrypted:");

      for(int j=1;j<=h;j++){
        for(i=0;i<p.length;i++){
          System.out.print(c[j][i]);
        }
        System.out.println("");
      }
    }
   }
```

Expno:10 Pseudo randam number(PSR)

Program:

```
import java.util.Random;

import java.util.Scanner;



public class psr {

    public static void main(String args[]){

        Scanner sc=new Scanner(System.in);

        Random rand=new Random(System.currentTimeMillis());

        int i;

        int count=0;

        System.out.println("enter the prime no:");

        int b=sc.nextInt();

        System.out.println("enter the value:");

        int v=sc.nextInt();

        for(i=1;i<=b;i++)

        {

            if(b%i==0)

            {

                count++;
            }
        }
        if(count==2){
            System.out.println(b+" is  prime number");
            System.out.println("");
            for(i=0;i<b;i++){
                System.out.println(rand.nextInt(v)+"\t");
            }
        }
        else{
            System.out.println(b+" is not a prime number");
        }
    }
}
```

Expno:11 Chinese Remainder Theorem(CRT)

Program:

```
import java.util.Random;
import java.util.Scanner;
```

```
public class psr {

    public static void main(String args[]){

        Scanner sc=new Scanner(System.in);

        Random rand=new Random(System.currentTimeMillis());

        int i;

        int count=0;

        System.out.println("enter the prime no:");

        int b=sc.nextInt();

        System.out.println("enter the value:");

        int v=sc.nextInt();

        for(i=1;i<=b;i++)

        {

            if(b%i==0)

            {

                count++;

            }

        }

        if(count==2){

            System.out.println(b+" is  prime number");

            System.out.println("");
            for(i=0;i<b;i++){
                System.out.println(rand.nextInt(v)+"\t");
            }
        }
        else{
            System.out.println(b+" is not a prime number");
        }
    }
}
```

Expno: 12 RSA Algorithm

Program:

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

class RSA
{
    public static void main(String args[])
    {
        Scanner sc=new Scanner(System.in);
        int p,q,n,z,d=0,e,i;
        System.out.println("Enter the number to be encrypted and decrypted");
        int msg=sc.nextInt();
        double c;
        BigInteger msgback;
        System.out.println("Enter 1st prime number p");
        p=sc.nextInt();
        System.out.println("Enter 2nd prime number q");
        q=sc.nextInt();

        n=p*q;
        z=(p-1)*(q-1);
        System.out.println("the value of z = "+z);

        for(e=2;e<z;e++)
        {
            if(gcd(e,z)==1)         // e is for public key exponent
            {
                break;
            }
        }
```

```
    System.out.println("the value of e = "+e);

    for(i=0;i<=9;i++)

    {

            int x=1+(i*z);

            if(x%e==0)      //d is for private key exponent

            {

                    d=x/e;

                    break;

            }

    }

    System.out.println("the value of d = "+d);

    c=(Math.pow(msg,e))%n;

    System.out.println("Encrypted message is : -");

    System.out.println(c);

//converting int value of n to BigInteger

    BigInteger N = BigInteger.valueOf(n);

    //converting float value of c to BigInteger

    BigInteger C = BigDecimal.valueOf(c).toBigInteger();

    msgback = (C.pow(d)).mod(N);

    System.out.println("Derypted message is : -");
```

```
                System.out.println(msgback);



        }



        static int gcd(int e, int z)

        {

                if(e==0)

                        return z;

                else

                        return gcd(z%e,e);

        }

}
```

Expno:13 Diffie Helman Key Enchange Algorithm

Program:

```
import java.util.*;


public class exp13diffi{

  public static void main(String args[]){

   Scanner sc=new Scanner(System.in);
```

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

    int q=sc.nextInt();

    System.out.println("Enter the alpha value");

    int halfa=sc.nextInt();

    System.out.println("Enter the XA value");

    int xa=sc.nextInt();

    System.out.println("Enter the XB value");

    int xb=sc.nextInt();

    long ya,yb;

    long s=1L;

    long s1=1L,s2=1L,s3=1L;

System.out.println("the given values are:");

System.out.println("halfa:"+halfa);

System.out.println("xa"+xa);

System.out.println("xb"+xb);

    for(int i=1;i<=xa;i++){

        s=halfa*s;

    }

     ya=s%q;

     System.out.println(ya+"="+halfa+"^"+xa+"mod"+q);
    for(int i=1;i<=xb;i++){
        s1=halfa*s1;

    }

     yb=s1%q;
     System.out.println(yb+"="+halfa+"^"+xb+"mod"+q);
    for(int i=1;i<=xa;i++){
        s2=s2*yb;
    }
  long k=s2%q;
  System.out.println(k+"="+yb+"^"+xa+"mod"+q);
   long de;
   for(int i=1;i<=xb;i++){
    s3=s3*ya;
}
de=s3%q;
System.out.println(de+"="+ya+"^"+xb+"mod"+q);
}
}
```

Expno:14 Secure Hash Algorithm (SHA)

Program:

```
import java.security.*;

class sha1

{

   public static void main(String[] a) {

      try {

         MessageDigest md = MessageDigest.getInstance("SHA1");

         System.out.println("Message digest object info:\n....");

         System.out.println("Algorithm=" + md.getAlgorithm());

         System.out.println("Provider=" + md.getProvider());

         System.out.println("ToString=" + md.toString());

         String input = "";

         md.update(input.getBytes());

         byte[]output = md.digest();

         System.out.println();

         System.out.println("SHA1(\""+input+"\") =" +bytesToHex(output));

         input = "abc";

         md.update(input.getBytes());

         output = md.digest();
```

```
        System.out.println();

        System.out.println("SHA1(\""+input+"\")="+bytesToHex(output));

                input="abcdefghijklmnopqrstuvwxyz";

        md.update(input.getBytes());

        output=md.digest();

        System.out.println();

        System.out.println("SHA1(\""+input+"\")="+bytesToHex(output));

        System.out.println();

    } catch (Exception e) {

        System.out.println("Exception:" + e);

    }

}

private static String bytesToHex(byte[]b)

{

    char hexDigit[]={'0','1','2','3','4','5','6','7','8','9','A','B','C','D','E','F'};

    StringBuffer buf=new StringBuffer();

    for(byte aB:b){

        buf.append(hexDigit[(aB>>4)&0x0f]);

        buf.append(hexDigit[aB&0x0f]);

    }
```

```
        return buf.toString();
    }
}
```

Expno:15 Digital Signature Standard

Program:

```
import java.security.KeyPair;
import java.security.KeyPairGenerator;
import java.security.PrivateKey;
import java.security.Signature;
import java.util.Scanner;
class exp15 dss {
public static void main(String args[]) throws Exception {
Scanner sc = new Scanner(System.in);
System.out.println("Enter some text");
String msg = sc.nextLine();
KeyPairGenerator keyPairGen = KeyPairGenerator.getInstance("DSA");
keyPairGen.initialize(2048);
KeyPair pair = keyPairGen.generateKeyPair();
PrivateKey privKey = pair.getPrivate();
Signature sign = Signature.getInstance("SHA256withDSA");
sign.initSign(privKey);
byte[] bytes = "msg".getBytes();
sign.update(bytes);
byte[] signature = sign.sign();
System.out.println("Digital signature for given text: "+new String(signature,
"UTF8"));
}}
```