**Name – Ganesh Halpatrao**

**Class - TYBSc CS Roll no 2096**

**Subject - INS Practical (sub 3 )**

INS Practical Doc

# Practical 1

Q Write program implement the following substitution cipher techniques

a)Caesar Cipher

b monoalphabetic Cipher

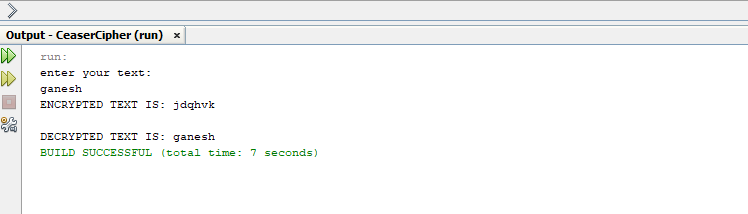
## A ) Caesar Cipher

One java application

Code

|  |
| --- |
| package CeaserCipher1;  import java.io.\*;    public class CeaserCipher  {  static String s;  static char ch;  public String encrypt(String str)  {  s ="";  for (int i=0;i<str.length();i++)  {  if (str.charAt(i)=='x')  s+='a';  else if (str.charAt(i)=='y')  s+='b';  else if(str.charAt(i)=='z')  s+='c';  else{  ch=str.charAt(i);  s+=(char)(ch+3);  }  }  return s;  }  public String decrypt(String str)  {  s="";  for(int i=0;i<str.length();i++)  {  if(str.charAt(i)=='a')  s+='x';  else if (str.charAt(i)=='b')  s+='y';  else if(str.charAt(i)=='c')  s+='z';  else  {  ch=str.charAt(i);  s+=(char)(ch-3);  }  }  return s;    }  public static void main(String[] args)throws Exception  {  CeaserCipher cc=new CeaserCipher();  String en;  BufferedReader br=new BufferedReader(new InputStreamReader(System.in));  System.out.println("enter your text:");  en=cc.encrypt(br.readLine().toLowerCase());  System.out.println("ENCRYPTED TEXT IS: "+en+"\n");  System.out.println("DECRYPTED TEXT IS: "+cc.decrypt(en));  }  } |

Output

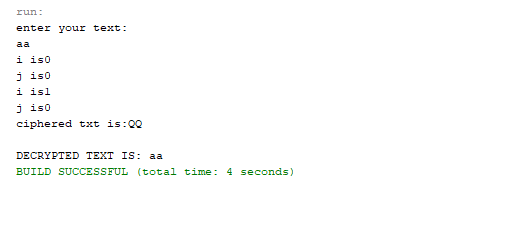


## B) Monalpahabetic

Code

|  |
| --- |
| package MonoAlphabetic;  import java.io.\*;  public class MonoAlphabetic  {  public static char plaintext[]={'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','\_'};  public static char charactertext[]={'Q','W','E','R','T','Y','U','T','O','P','A','S','D','F','G','H','J','K','L','Z','X','C','V','B','N','M','^','\*'};  public static String doEncryption(String s)  {  char ciphertext[]=new char[(s.length())];  for(int i=0;i<s.length();i++)  {  for(int j=0;j<28;j++)  {  System.out.println("i is"+i);  System.out.println("j is"+j);  if (plaintext[j]==s.charAt(i))  {  ciphertext[i]=charactertext[j];  break;  }  }  }  return(new String(ciphertext));  }  public static String doDecryption(String s)  {  char newplaintext[]=new char[(s.length())];  for(int i=0;i<s.length();i++)  {  for(int j=0;j<28;j++)  {  if(charactertext[j]==s.charAt(i))    {  newplaintext[i]=plaintext[j];  break;  }  }  }  return(new String(newplaintext));  }  public static void main(String[] args)throws IOException  {  MonoAlphabetic ma=new MonoAlphabetic();  BufferedReader br=new BufferedReader(new InputStreamReader(System.in));  System.out.println("enter your text:");  String in=ma.doEncryption(br.readLine().toLowerCase());  System.out.println("ciphered txt is:"+in+"\n");  System.out.println("DECRYPTED TEXT IS: "+ma.doDecryption(in));  }  } |

Output



# Practical 2

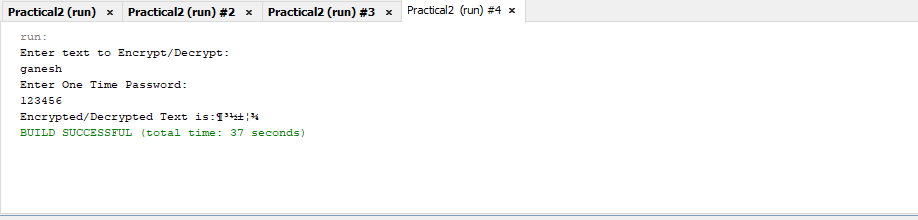
Q Write program implement the following substitution cipher techniques

## Vernam Cipher

Code

|  |
| --- |
| import java.util.Scanner;  public class VernamCipher {  public static void main(String[]args)  {  String text,key,output="";  char t,k;  int x;  Scanner s=new Scanner(System.in);  System.out.println("Enter text to Encrypt/Decrypt: ");  text=s.nextLine();  System.out.println("Enter One Time Password: ");  key=s.nextLine();  for(int i=0;i<text.length();i++){  t=text.charAt(i);  k=key.charAt(i);  x=t^k;  output+=(char)(x+96);  }  System.out.println("Encrypted/Decrypted Text is:"+output);  }  } |

Output



# Practical 3

Q Write program implement the following substitution cipher techniques

a)Rail Fence Cipher

b) Simple Columnar Technique

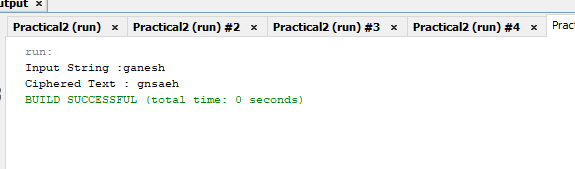
Solution

## A ) Rail Fence Cipher

Code

|  |
| --- |
| import java.io.\*;  public class RailFence {  public static void main(String[]args)  {  String input="ganesh";  String output="";  int len =input.length();  System.out.println("Input String :"+input);  for(int i=0;i<len;i+=2){  output+=input.charAt(i);  }  for(int i=1;i<len;i+=2){  output+=input.charAt(i);  }  System.out.println("Ciphered Text : "+ output);  }  } |

Output



## B) Simple Columnar Technique

Code

|  |
| --- |
| import java.io.\*;  public class SCTS {  public static void main(String[] arges)throws IOException{  BufferedReader br= new BufferedReader(new InputStreamReader(System.in));  System.out.println("Enter yor Plain Text: ");  String accept =br.readLine();  System.out.println("Enter no. of rows:");  int r= Integer.parseInt(br.readLine());  System.out.println("Enter no. of columns:");  int c=Integer.parseInt(br.readLine());  int count=0;  char cont[][]=new char[r][c];  for(int i=0;i<r;i++){  for (int j=0;j<c;j++){  if(count>=accept.length()){  cont[i][j]=' ';  count++;  }  else{  cont[i][j]=accept.charAt(count);  count++;  }  }  }  System.out.println("\nEnter the order of choices: ");  int choice[]=new int[c];  for(int k=0;k<c;k++){  System.out.println("Choice"+k+"-->");  choice[k]=Integer.parseInt(br.readLine());  }  System.out.println("\nCipher Text in matrix is-->");  String cipher=" ";  for(int j=0;j<c;j++){  int k=choice[j];  for(int i=0;i<r;i++){  cipher+=cont[i][k];  }  }  cipher=cipher.trim();  System.out.println(cipher);  }  } |

Output



# Practical 4

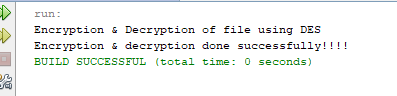
Q Write program to encrypt and decrypt strings Using

## A) DES algorithm

Code

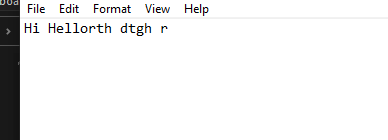
|  |
| --- |
| import java.io.\*;  import javax.crypto.\*;  public class DES {  Cipher ecipher,dcipher;  byte[] buf=new byte[1024];  public DES(SecretKey key)  {  try  {  ecipher=Cipher.getInstance("DES");  ecipher.init(Cipher.ENCRYPT\_MODE,key);  dcipher=Cipher.getInstance("DES");  dcipher.init(Cipher.DECRYPT\_MODE,key);  }  catch(Exception e)  {  System.out.println("Exception occur:" + e);  }  }  public void encrypt(InputStream in,OutputStream out)  {  try{  int numRead=0;  out=new CipherOutputStream(out,ecipher);  while((numRead=in.read(buf))>=0)  {  out.write(buf,0,numRead);  }  out.close();  }  catch(Exception e)  {  System.out.println("Exception occur:" + e);  }  }  public void decrypt(InputStream in, OutputStream out)  {  try  {  int numRead=0;  in=new CipherInputStream(in,dcipher);  while((numRead=in.read(buf))>=0)  {  out.write(buf,0,numRead);  }  out.close();  }  catch(Exception e)  {  System.out.println("Exception occur: " + e);  }  }  /\*\*  \* @param args the command line arguments  \*/  public static void main(String[] args){  try  {  System.out.println("Encryption & Decryption of file using DES");  SecretKey key=KeyGenerator.getInstance("DES").generateKey();  DES encrypter=new DES(key);  encrypter.encrypt(new FileInputStream("D:\\ty\\2096\\INS\\pract 4\\practicalINS4\\src\\Test.txt"),new FileOutputStream("ciphertext.txt"));  encrypter.decrypt(new FileInputStream("ciphertext.txt"),new FileOutputStream("text2.txt"));  System.out.println("Encryption & decryption done successfully!!!!");  }  catch(Exception e)  {  System.out.println("Exception occur:"+e);  }      }    } |

Output

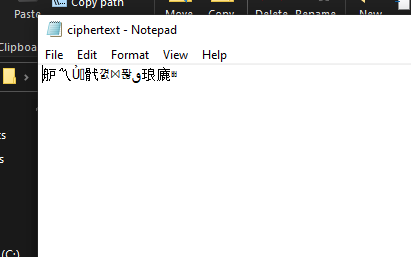


And two file are created

Text2



Ciphertext



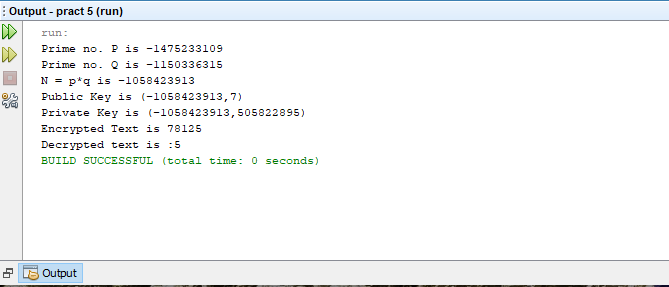
# Practical 5

Write program to implement RSA Algorithm to perform encryption / decryption of a given string.

## RSA

|  |
| --- |
| package rsa;  import java.math.BigInteger;  import java.security.SecureRandom;  public class RSA {  SecureRandom r;  BigInteger p, q, p1, q1, n, n1, e, d, msg, ct, pt;  public RSA() {  int bitLength = 512;  int certainty = 100;  r = new SecureRandom();  // step:1 Generate prime no. p and Q  p = new BigInteger(bitLength,certainty,r );  q = new BigInteger(bitLength, certainty, r);  // step:2 n = p\*q  n = p.multiply(q);  System.out.println("Prime no. P is "+p.intValue());  System.out.println("Prime no. Q is "+q.intValue());  System.out.println("N = p\*q is "+n.intValue());  // step:3 Generating public key(E)  p1 = p.subtract(new BigInteger("1"));  q1 = q.subtract(new BigInteger("1"));  n1 = p1.multiply(q1);  e = new BigInteger("2");  while(n1.gcd(e).intValue() > 1 || e.compareTo(p1) != -1)  e = e.add(new BigInteger("1"));  System.out.println("Public Key is ("+ n.intValue()+","+e.intValue()+")");  d = e.modInverse(n1);  System.out.println("Private Key is ("+ n.intValue()+","+d.intValue()+")");  // Step:5 Encryption CT = (PT)^e mod n  msg = new BigInteger("5");  ct = encrypt();  System.out.println("Encrypted Text is "+ ct.intValue());  // step:6 Decryption PT= (CT) ^d mod n  pt = decrypt(ct);  System.out.println("Decrypted text is :"+ pt.intValue());  }  public BigInteger encrypt() {  return msg.modPow(e, n);  }  public BigInteger decrypt(BigInteger ct) {  return ct.modPow(d, n);  }  public static void main(String[] args) {  new RSA();  }  } |

Output



# Practical 6

Q )

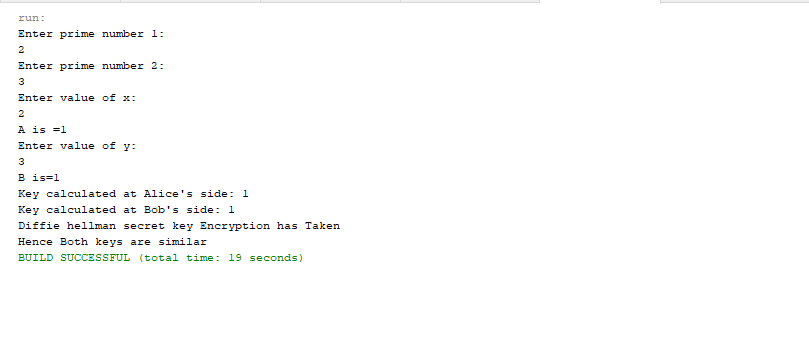
Write a program to implement the Diffie -Hellman Key Agreement Algorithm to generate symmetric Keys .

## Diffie -Hellman

Solution Code

|  |
| --- |
| import java.io.\*;  import java.math.BigInteger;  public class Diffie {  public static void main(String[]args)throws IOException{  BufferedReader br = new BufferedReader(new InputStreamReader(System.in));  System.out.println("Enter prime number 1:");  BigInteger p=new BigInteger(br.readLine());  System.out.println("Enter prime number 2:");  BigInteger g=new BigInteger(br.readLine());  System.out.println("Enter value of x:");  BigInteger x=new BigInteger(br.readLine());  BigInteger A=g.modPow(x,p);  System.out.println("A is ="+A);  System.out.println("Enter value of y:");  BigInteger y=new BigInteger(br.readLine());  BigInteger B=g.modPow(y,p);  System.out.println("B is="+B);  BigInteger k1=B.modPow(x,p);  System.out.println("Key calculated at Alice's side: "+k1);  BigInteger k2=A.modPow(y,p);  System.out.println("Key calculated at Bob's side: "+k2);  System.out.println("Diffie hellman secret key Encryption has Taken");  if(k1.equals(k2)){  System.out.println("Hence Both keys are similar");  }  else  System.out.println("Hence keys are not similar");    }    } |

Output

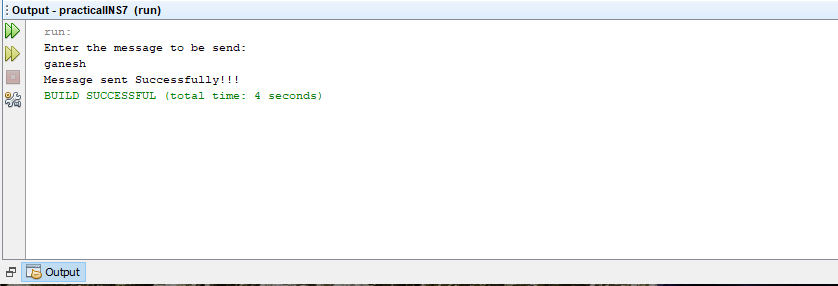


# Practical 7

A) Write program to implement the MD5 algorithm compute the message digest.

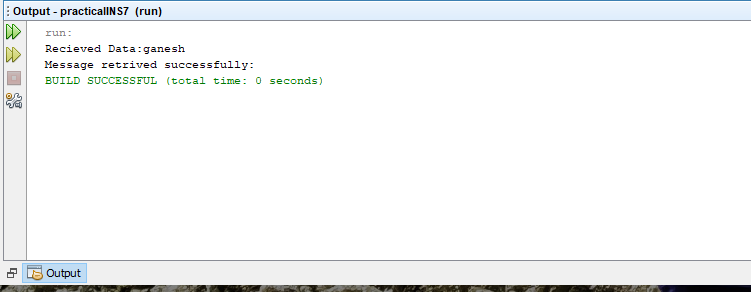
## Send

|  |
| --- |
| import java.io.\*;  import java.security.\*;  public class send {  public static void main(String args[])  {  String input;  byte buffer[]=new byte[1024];  System.out.println("Enter the message to be send: ");  try{  BufferedReader br= new BufferedReader(new InputStreamReader(System.in));  input=br.readLine();  FileOutputStream fos= new FileOutputStream("D:\\ty\\2096\\INS\\pract 7\\JavaApplication16\\Text.txt");  ObjectOutputStream oos= new ObjectOutputStream(fos);  MessageDigest md = MessageDigest.getInstance("MD5");  buffer=input.getBytes();  md.update(buffer);  oos.writeObject(input);//original data  oos.writeObject(md.digest());//fingerprint of data  System.out.println("Message sent Successfully!!!");  }  catch(Exception e)  {  e.printStackTrace();  }  }  } |



## Receive

|  |
| --- |
| import java.io.\*;  import java.security.\*;  public class Receive {  public static void main(String args[])  {  byte dig[] =new byte[1024];  try{  FileInputStream fis = new FileInputStream("D:\\ty\\2096\\INS\\pract 7\\JavaApplication16\\Text.txt");  ObjectInputStream ois = new ObjectInputStream(fis);  Object obj= ois.readObject();  String data =(String)obj;  System.out.println("Recieved Data:"+data);  obj= ois.readObject();  dig=(byte[])obj;  MessageDigest md= MessageDigest.getInstance("MD5");  md.update(data.getBytes());  if(MessageDigest.isEqual(md.digest(),dig))  System.out.println("Message retrived successfully:");  ois.close();  }  catch(StreamCorruptedException e){  System.out.println("Message is corrupted!!!");  }  catch(IOException e){  }  catch(NoSuchAlgorithmException e){  e.printStackTrace();  }  catch(ClassNotFoundException e){  e.printStackTrace();  }  }  } |



# Practical 8

Q Write program to encrypt and decrypt strings using AES algorithm

## A) ASE

|  |
| --- |
| import java.io.\*;  import javax.crypto.\*;  public class AES {  Cipher ecipher,dcipher;  byte[] buf=new byte[1024];  public AES(SecretKey key)  {  try  {  ecipher=Cipher.getInstance("DES");  ecipher.init(Cipher.ENCRYPT\_MODE,key);  dcipher=Cipher.getInstance("DES");  dcipher.init(Cipher.DECRYPT\_MODE,key);  }  catch(Exception e)  {  System.out.println("Exception occur:" + e);  }  }  public void encrypt(InputStream in,OutputStream out)  {  try{  int numRead=0;  out=new CipherOutputStream(out,ecipher);  while((numRead=in.read(buf))>=0)  {  out.write(buf,0,numRead);  }  out.close();  }  catch(Exception e)  {  System.out.println("Exception occur:" + e);  }  }  public void decrypt(InputStream in, OutputStream out)  {  try  {  int numRead=0;  in=new CipherInputStream(in,dcipher);  while((numRead=in.read(buf))>=0)  {  out.write(buf,0,numRead);  }  out.close();  }  catch(Exception e)  {  System.out.println("Exception occur: " + e);  }  }  /\*\*  \* @param args the command line arguments  \*/  public static void main(String[] args){  try  {  System.out.println("Encryption & Decryption of file using DES");  SecretKey key=KeyGenerator.getInstance("DES").generateKey();  AES encrypter=new AES(key);  encrypter.encrypt(new FileInputStream("D:\\ty\\2096\\INS\\pract 4\\practicalINS4\\src\\Test.txt"),new FileOutputStream("ciphertext.txt"));  encrypter.decrypt(new FileInputStream("ciphertext.txt"),new FileOutputStream("text2.txt"));  System.out.println("Encryption & decryption done successfully!!!!");  }  catch(Exception e)  {  System.out.println("Exception occur:"+e);  }      }    } |

Same output as des