Date:11/09/2020

**Practical no 4**

**AIM:** Write program to encrypt and decrypt strings using

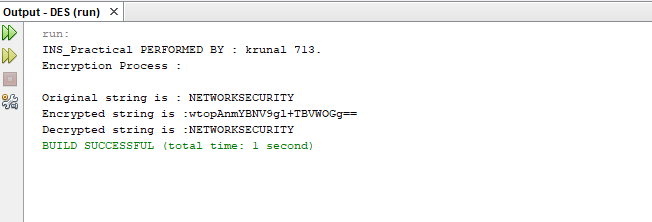
1) DES Algorithm 2) AES Algorithm

**CODE**

**1) DES Algorithm**

|  |
| --- |
| import java.util.logging.Level;  import java.util.logging.Logger;  import java.util.Base64;  import javax.crypto.Cipher;  import javax.crypto.KeyGenerator;  import javax.crypto.SecretKey;  public class DES {  public static SecretKey getSecretEncryptionKey() throws Exception{  KeyGenerator generator=KeyGenerator.getInstance("DES");  SecretKey secKey=generator.generateKey();  return secKey;  }  public String encrypt(SecretKey key,String Plaintext) throws Exception{  byte[] utf8=Plaintext.getBytes();  Cipher ecipher=Cipher.getInstance("DES");  ecipher.init(Cipher.ENCRYPT\_MODE, key);  byte[] enc=ecipher.doFinal(utf8);  Base64.Encoder encoder=Base64.getEncoder();  String et=encoder.encodeToString(enc);  return et;  }  public String decrypt(SecretKey key,String Ciphertext) throws Exception{  Base64.Decoder decoder = Base64.getDecoder();  byte[] dec=decoder.decode(Ciphertext);  Cipher dcipher=Cipher.getInstance("DES");  dcipher.init(Cipher.DECRYPT\_MODE, key);  byte[] utf8=dcipher.doFinal(dec);  return new String(utf8,"UTF8");    }  public static void main(String[] args){  try{  System.out.println("INS\_Practical PERFORMED BY : krunal 713.");  System.out.println("----'--Encrypting string using DES--'----");  System.out.println();  String message ="NETWORKSECURITY";  DES d=new DES();  SecretKey key=getSecretEncryptionKey();  String Encrypted=d.encrypt(key, message);  String Decrypted=d.decrypt(key, Encrypted);  System.out.println("Original String is : "+ message);  System.out.println("Encrypted String is : "+ Encrypted);  System.out.println("Decrypted String is : "+ Decrypted);  }catch (Exception ex){  Logger.getLogger(DES.class.getName()).log(Level.SEVERE,null,ex);  }  }  } |

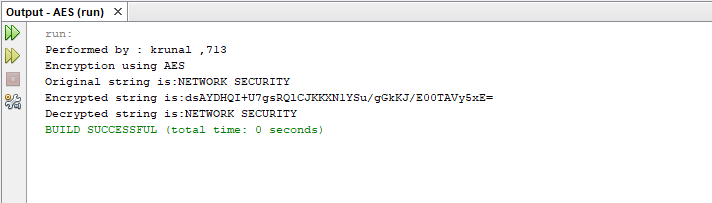
**Output:**

****

**b) AES CODE**

|  |
| --- |
| package aes;  import java.util.logging.Logger;  import java.util.logging.Level;  import javax.crypto.Cipher;  import javax.crypto.KeyGenerator;  import javax.crypto.SecretKey;  public class AES {  public static SecretKey getSecretEncryptionKey() throws Exception{  KeyGenerator generator = KeyGenerator.getInstance("AES");  generator.init(128);  SecretKey secKey= generator.generateKey();  return secKey;  }    public String encrypt(SecretKey key,String Plaintext)throws Exception{  byte[] utf8= Plaintext.getBytes("UTF8");  Cipher ecipher= Cipher.getInstance("AES");  ecipher.init(Cipher.ENCRYPT\_MODE,key);  byte[] enc= ecipher.doFinal(utf8);  return new sun.misc.BASE64Encoder().encode(enc);  }    public String decrypt(SecretKey key,String Ciphertext) throws Exception{  byte[] dec= new sun.misc.BASE64Decoder().decodeBuffer(Ciphertext);  Cipher dcipher= Cipher.getInstance("AES");  dcipher.init(Cipher.DECRYPT\_MODE,key);  byte[] utf8= dcipher.doFinal(dec);  return new String(utf8, "UTF8");  }    public static void main (String[]args) throws Exception  {  try{  System.out.println("Performed by : krunal ,713");  System.out.println("Encryption using AES");  String message="NETWORK SECURITY";  AES d= new AES();  SecretKey key= getSecretEncryptionKey();  String Encrypted= d.encrypt(key, message);  String decrypted = d.decrypt(key,Encrypted);  System.out.println("Original string is:" +message);  System.out.println("Encrypted string is:" + Encrypted);  System.out.println("Decrypted string is:" +decrypted);  }  catch(Exception ex){  Logger.getLogger(AES.class.getName()).log(Level.SEVERE,null,ex) ;  }  }  } |

**Output:**



Date:29/09/2020

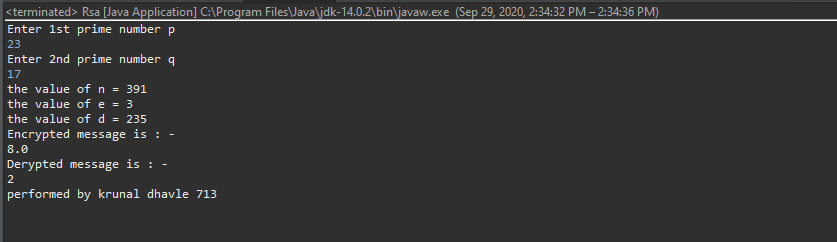
**Practical no 5**

**AIM:** Write a program to implement RSA algorithm to perform encryption / decryption of a given string.

**CODE**

|  |
| --- |
| package prac5;  import java.util.\*;  import java.math.\*;  public class Rsa {  public static void main(String[] args) {  // TODO Auto-generated method stub  Scanner sc=new Scanner(System.in);  int p,q,n,z,d=0,e,i;  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();  sc.close();  n=p\*q;  z=(p-1)\*(q-1);  System.out.println("the value of n = "+n);  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(2,e))%n;    System.out.println("Encrypted message is : -");  System.out.println(c);    BigInteger N = BigInteger.valueOf(n);    BigInteger C = BigDecimal.valueOf(c).toBigInteger();  msgback = (C.pow(d)).mod(N);    System.out.println("Derypted message is : -");  System.out.println(msgback);  System.out.println("performed by krunal dhavle 713");  }  static int gcd(int e, int z)  {  if(e==0)  return z;  else  return gcd(z%e,e);  }  } |

**Output:**



Date:06/10/2020

**Practical no 6**

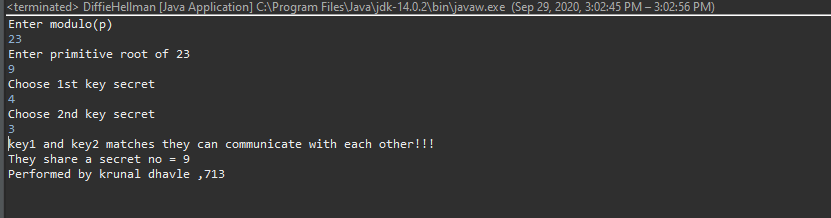
**AIM:** Write a program to implement the Diffie-Hellman Key Agreement algorithm to generate symmetric keys.

**CODE:-**

**Method 1:-**

|  |
| --- |
| package prac6;  import java.util.\*;  public class DiffieHellman {  public static void main(String[] args) {  // TODO Auto-generated method stub  Scanner sc=new Scanner(System.in);  System.out.println("Enter modulo(p)");  int p=sc.nextInt();  System.out.println("Enter primitive root of "+p);  int g=sc.nextInt();  System.out.println("Choose 1st key secret");  int a=sc.nextInt();  System.out.println("Choose 2nd key secret");  int b=sc.nextInt();  sc.close();  int A = (int)Math.pow(g,a)%p;  int B = (int)Math.pow(g,b)%p;    int S\_A = (int)Math.pow(B,a)%p;  int S\_B =(int)Math.pow(A,b)%p;    if(S\_A==S\_B)  {  System.out.println("key1 and key2 matches they can communicate with each other!!!");  System.out.println("They share a secret no = "+S\_A);  System.out.println("Performed by krunal dhavle ,713");  }    else  {  System.out.println("key1 and key2 matches they cannot communicate with each other!!!");  System.out.println("Performed by krunal dhavle ,713");  }  }} |

**Output:**



**Method 2 :-**

**Bob.java**

|  |
| --- |
| package prac6;  import java.io.\*;  import java.net.ServerSocket;  import java.net.Socket;  import java.util.Scanner;  public class Bob {  public static void main(String[] args) throws IOException {  ServerSocket ss = new ServerSocket(5000);  Socket s = ss.accept();  DataInputStream in = new DataInputStream(s.getInputStream());  int n = in.readInt();  int g = in.readInt();  Scanner sc = new Scanner(System.in);  System.out.println("Enter the value of y");  int y = sc.nextInt();  System.out.println("n=" +n);  System.out.println("g=" +g);  int d =(int)Math.pow(g, y);  int B =d%n;  System.out.println("The calculated value of B is " +B);  System.out.println("bob sends the value of B " +B+ " to alice");  int A = in.readInt();  int b = (int)Math.pow(A,y);  double K2 = b%n;  System.out.println("the calculated value of k2 is " +K2);  DataOutputStream out = new DataOutputStream(s.getOutputStream());  out.writeInt(B);  System.out.println("performed by krunal 713");  }  } |

**Alice.java**

|  |
| --- |
| package prac6;  import java.io.\*;  import java.net.Socket;  import java.util.Scanner;  public class Alice {  public static void main(String[] args) throws IOException {  Socket cs = new Socket("localhost" ,5000);  Scanner sc = new Scanner(System.in);  System.out.println("Enter the value of n and g ");  int n = sc.nextInt();  int g = sc.nextInt();  System.out.println("n=" +n);  System.out.println("g=" +g);  DataOutputStream out = new DataOutputStream(cs.getOutputStream());  out.writeInt(n);  out.writeInt(g);  System.out.println("Enter the value of x : ");  int x = sc.nextInt();  int c =(int)Math.pow(g,x);  int A = c%n;  System.out.println("the calculated value of A is " +A);  out.writeInt(A);  System.out.println("Alice sends the value of a " +A + "to bob");  DataInputStream in = new DataInputStream(cs.getInputStream());  int B = in.readInt();  int a = (int)Math.pow(B, x);  double K1 = a % n;  System.out.println("the calculated value for k1 is " +K1);  System.out.println("performed by krunal 713");  }  } |

**Output:**

