**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);  }  } |

