**import java.util.\*;**

**import java.math.\*;**

**public class RSA**

**{**  **static** BigInteger p,q,e,d,n,phi;

**static** int **bitLength=256;**

**static** Scanner **S=new Scanner(System.in);**

**static** Random **R=new Random();**

**public** static void main **(String args[])**

**{** p=BigInteger.probablePrime(bitLength,R);

q=BigInteger.probablePrime(bitLength,R);

n=p.multiply(q);

e=BigInteger.probablePrime(bitLength/2,R);

phi=p.subtract(BigInteger.ONE).multiply(q.subtract(BigInteger.ONE)); while(phi.gcd(e).compareTo(BigInteger.ONE)!=0 && e.compareTo(phi)<0)

e.add(BigInteger.ONE);

d=e.modInverse(phi);

String msg="";

System.out.print("Enter The Msg : ");

msg=S.nextLine();

byte msg\_arr[]=msg.getBytes();

System.out.println("Msg Byte Array : "+display(msg\_arr));

byte en[]=encrypt(msg\_arr);

System.out.println("Encrypted Byte Array : "+display(en));

byte de[]=decrypt(en);

System.out.println("Decrypted Byte Array : "+display(de));

System.out.println("Received Msg : "+ new String(de)); }

static byte[] encrypt(byte a[])

{ return (new BigInteger(a).modPow(e,n)).toByteArray(); }

static byte[] decrypt(byte a[])

{ return (new BigInteger(a).modPow(d,n)).toByteArray(); }

static String display(byte a[])

{String s="";

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

s+=Byte.toString(a[i]);

return s; } }

**Output :**Enter The Msg : This is a sample

Msg Byte Array:841041051153210511532973211597109112108101

Encrypted Byte Array:7-38-64-487597-725231-45-87-6981-29-17-73-34127-101108-1289-126-769143-126-56-22- 21-27-7819120852868-91-81-47-105-7937-75-48-10681-6651-43-74-126-28-10468-853610941-38-58-127-126- 10910936-63347-69127

Decrypted Byte Array:841041051153210511532973211597109112108101 Received Msg: This is a sample