Ex No : 2 A

DES ALGORITHM

**Program :**

import java.io.\*;

import javax.crypto.Cipher;

import javax.crypto.KeyGenerator;

import javax.crypto.SecretKey;

class DesEncrypter {

Cipher ecipher;

Cipher dcipher;

DesEncrypter(SecretKey key) throws Exception {

ecipher = Cipher.getInstance("DES");

dcipher = Cipher.getInstance("DES");

ecipher.init(Cipher.ENCRYPT\_MODE, key);

dcipher.init(Cipher.DECRYPT\_MODE, key);

}

public String encrypt(String str) throws Exception {

// Encode the string into bytes using utf-8

byte[] utf8 = str.getBytes("UTF8");

// Encrypt

byte[] enc = ecipher.doFinal(utf8);

// Encode bytes to base64 to get a string

return new sun.misc.BASE64Encoder().encode(enc);

}

public String decrypt(String str) throws Exception {

// Decode base64 to get bytes

byte[] dec = new sun.misc.BASE64Decoder().decodeBuffer(str);

byte[] utf8 = dcipher.doFinal(dec);

// Decode using utf-8

return new String(utf8, "UTF8");

}

}

public class Main {

static BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

public static void main(String[] argv) throws Exception {

SecretKey key = KeyGenerator.getInstance("DES").generateKey();

DesEncrypter encrypter = new DesEncrypter(key);

System.out.print("Enter the Plain Text: ");

String msg=br.readLine();

String encrypted = encrypter.encrypt(msg);

String decrypted = encrypter.decrypt(encrypted);

System.out.println("Encrypted Text: "+encrypted);

System.out.println("Decrypted Text: "+decrypted);

}

}

**Output :**



Ex No : 2 B

RSA ALGORITHM

**Program :**

import java.io.BufferedReader;

import java.io.InputStreamReader;

import java.math.\*;

import java.util.Random;

import java.util.Scanner;

public class RSA {

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

public static void main(String[] args) {

// TODO code application logic here

System.out.print("Enter a Prime number: ");

BigInteger p = sc.nextBigInteger(); // Here's one prime number..

System.out.print("Enter another prime number: ");

BigInteger q = sc.nextBigInteger(); // ..and another.

BigInteger n = p.multiply(q);

BigInteger n2 = p.subtract(BigInteger.ONE).multiply(q.subtract(BigInteger.ONE));

BigInteger e = generateE(n2);

BigInteger d = e.modInverse(n2); // Here's the multiplicative inverse

System.out.println("Encryption keys are: " + e + ", " + n);

System.out.println("Decryption keys are: " + d + ", " + n);

}

public static BigInteger generateE(BigInteger fiofn) {

int y, intGCD;

BigInteger e;

BigInteger gcd;

Random x = new Random();

do {

y = x.nextInt(fiofn.intValue()-1);

String z = Integer.toString(y);

e = new BigInteger(z);

gcd = fiofn.gcd(e);

intGCD = gcd.intValue();

}

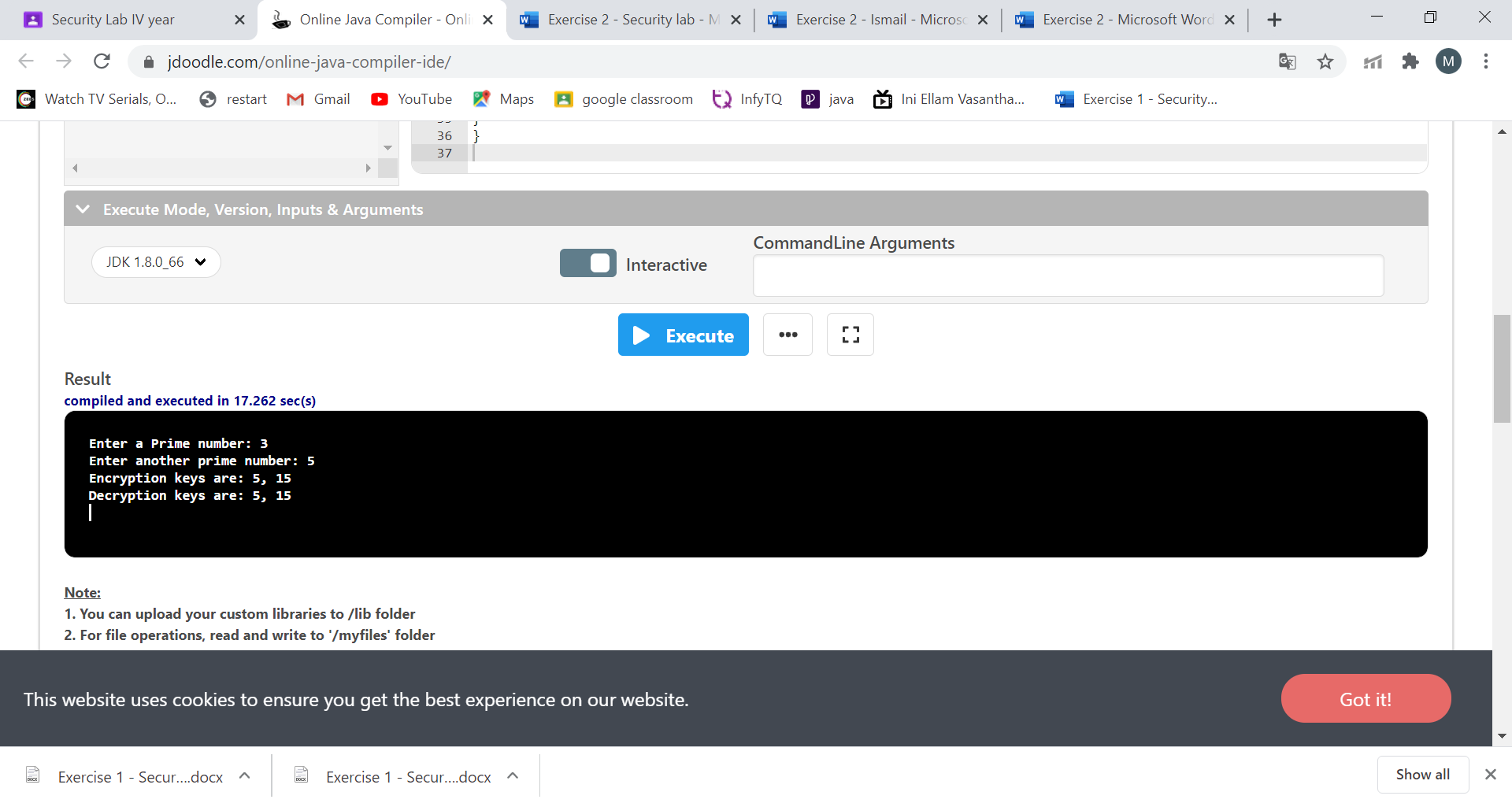
while(y <= 2 || intGCD != 1);

return e;

}

}

**Output :**



Ex No : 2 C

DIFFIE HELLMAN ALGORITHM

**Program :**

import java.io.\*;

import java.math.BigInteger;

class Diffie

{

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

{

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

System.out.print("Enter prime number: ");

BigInteger p=new BigInteger(br.readLine());

System.out.print("Enter primitive root of "+p+": ");

BigInteger g=new BigInteger(br.readLine());

System.out.print("Enter value for x less than "+p+": ");

BigInteger x=new BigInteger(br.readLine());

BigInteger R1=g.modPow(x,p);

System.out.println("R1="+R1);

System.out.print("Enter value for y less than "+p+": ");

BigInteger y=new BigInteger(br.readLine());

BigInteger R2=g.modPow(y,p);

System.out.println("R2="+R2);

BigInteger k1=R2.modPow(x,p);

System.out.println("Key calculated at Alice's side: "+k1);

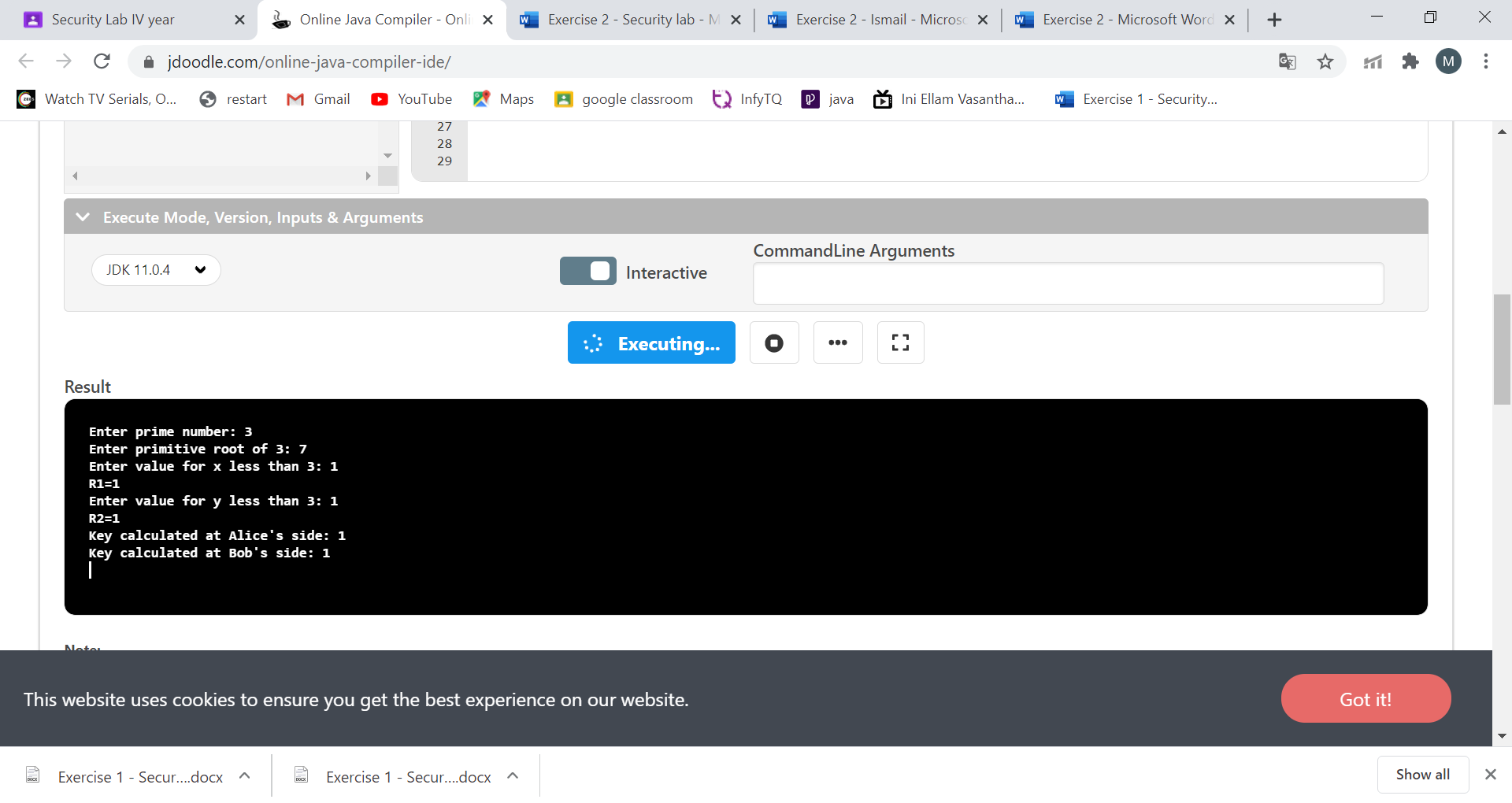
BigInteger k2=R1.modPow(y,p);

System.out.println("Key calculated at Bob's side: "+k2);

}

}

**Output :**



Ex No : 2 D

MD5 ALGORITHM

**Program :**

import java.security.\*;

public class MD5 {

public static void main(String[] a) {

try {

MessageDigest md = MessageDigest.getInstance("MD5");

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

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("MD5(\""+input+"\") = " +bytesToHex(output));

input = "abc";

md.update(input.getBytes());

output = md.digest();

System.out.println();

System.out.println("MD5(\""+input+"\") = " +bytesToHex(output));

input = "abcdefghijklmnopqrstuvwxyz";

md.update(input.getBytes());

output = md.digest();

System.out.println();

System.out.println("MD5(\"" +input+"\") = " +bytesToHex(output));

System.out.println("");

}

catch (Exception e) {

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

}

}

public 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 (int j=0; j<b.length; j++) {

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

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

}

return buf.toString();

}

}

**Output :**



Ex No : 2 E

SHA1 ALGORITHM

**Program :**

import java.security.\*;

public class SHA1 {

public static void main(String[] a) {

try {

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

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

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);

}

}

public 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 (int j=0; j<b.length; j++) {

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

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

}

return buf.toString();

}

}

**Output :**

