**KESHAV MEMORIAL INSTITUTE OF TECHNOLOGY**

**(KMIT)**

**(Approved by AICTE, New Delhi and Affiliated to JNTUH)**

**Narayanaguda, Hyderabad – 500029**

****

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

**LAB MANUAL**

**CRYPTOGRAPHY AND NETWORK SECURITY LAB**

**B.Tech IV YEAR I SEM (R18 JNTUH REGULATIONS)**

**ACADEMIC YEAR 2021-22**

**INDEX**

| **S.NO** | **CONTENTS** | **PAGE NO** |
| --- | --- | --- |
| I | V/M /PEOS/POS/PSOS | 2 |
| II | Syllabus | 5 |
| III | Course outcomes | 6 |
| IV | List of Experiments | 7 |
| V | Experiments | 8 |
| **Exp No** | **Experiment Name** |  |
| 1 | Write a C program that contains a string (char pointer) with a value ‘Hello world’. The program should XOR each character in this string with 0 and displays the result. | 8 |
| 2 | Write a C program that contains a string (char pointer) with a value ‘Hello world’. The program should AND or and XOR each character in this string with 127 and display the result. | 9 |
| 3 | Write a Java program to perform encryption and decryption using the following Algorithms:  a. Ceaser cipher  b. Substitution cipher  c. Hill Cipher | 11 |
| 4 | Write a C/JAVA program to implement the DES algorithm logic. | 19 |
| 5 | Write a C/JAVA program to implement the Blowfish algorithm logic. | 22 |
| 6 | Write a C/JAVA program to implement the Rijndael algorithm logic. | 24 |
| 7 | Write the RC4 logic in Java Using Java cryptography; encrypt the text “Hello world” using Blowfish. Create your own key using Java key tool. | 26 |
| 8 | Write a Java program to implement RSA algorithm. | 29 |
| 9 | Implement the Diffie-Hellman Key Exchange mechanism using HTML and JavaScript. | 32 |
| 10 | Calculate the message digest of a text using the SHA-1 algorithm in JAVA. | 35 |
| 11 | Calculate the message digest of a text using the MD5 algorithm in JAVA. | 37 |



**Vision of the Institution:**

* To be the fountain head in producing highly skilled, globally competent engineers.
* Producing quality graduates trained in the latest software technologies and related tools and striving to make India a world leader in software products and services.

**Mission of the Institution:**

* To provide a learning environment that inculcates problem solving skills, professional, ethical responsibilities, lifelong learning through multi modal platforms and prepare students to become successful professionals.
* To establish Industry Institute Interaction to make students ready for the industry.
* To provide exposure to students on latest hardware and software tools.
* To promote research based projects/activities in the emerging areas of technology convergence.
* To encourage and enable students to not merely seek jobs from the industry but also to create new enterprises
* To induce a spirit of nationalism which will enable the student to develop, understand India’s challenges and to encourage them to develop effective solutions.
* To support the faculty to accelerate their learning curve to deliver excellent service to students



**Department of Computer Science & Engineering**

**Vision of the Department**:

* To be among the region's premier teaching and research Computer Science and Engineering departments producing globally competent and socially responsible graduates in the most conducive academic environment.

**Mission of the Department**:

* To provide faculty with state of the art facilities for continuous professional development and research, both in foundational aspects and of relevance to emerging computing trends.
* To impart skills that transform students to develop technical solutions for societal needs and inculcate entrepreneurial talents.
* To inculcate an ability in students to pursue the advancement of knowledge in various specializations of Computer Science and Engineering and make them industry-ready.
* To engage in collaborative research with academia and industry and generate adequate resources for research activities for seamless transfer of knowledge resulting in sponsored projects and consultancy.
* To cultivate responsibility through sharing of knowledge and innovative computing solutions that benefit the society-at-large.
* To collaborate with academia, industry and community to set high standards in academic excellence and in fulfilling societal responsibilities.

**PROGRAM    EDUCATIONAL    OBJECTIVES:**

* Graduates will have successful careers in computer related engineering fields or will be able to successfully pursue advanced higher education degrees.
* Graduates will try and provide solutions to challenging problems in their profession by applying Information Technology principles.
* Graduates will engage in life-long learning and professional development by rapidly adapting changing work environment.
* Graduates will communicate effectively, work collaboratively and exhibit high levels of professionalism and ethical responsibility

**Computer Science & Engineering**

| **PO1** | **Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, andan engineering specialization to the solution of complex engineering problems. |
| --- | --- |
| **PO2** | **Problem analysis**: Identify, formulate, review research literature, and analyze complex engineeringproblems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences. |
| **PO3** | **Design/development of solutions**: Design solutions for complex engineering problems and designsystem components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations. |
| **PO4** | **Conduct investigations of complex problems**: Use research-based knowledge and research methodsincluding design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions. |
| **PO5** | **Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modernengineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations |
| **PO6** | **The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal,health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering pra |
| **PO7** | **Environment and sustainability**: Understand the impact of the professional engineering solutions insocietal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development. |
| **PO8** | **Ethics**: Apply ethical principles and commit to professional ethics and responsibilities and norms of theengineering practice. |
| **PO9** | **Individual and team work**: Function effectively as an individual, and as a member or leader in diverseteams, and in multidisciplinary settings. |
| **PO10** | **Communication**: Communicate effectively on complex engineering activities with the engineeringcommunity and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions |
| **P011** | **Project management and finance**: Demonstrate knowledge and understanding of the engineering andmanagement principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments. |
| **P012** | **Life-long learning**: Recognize the need for, and have the preparation and ability to engage inindependent and life-long learning in the broadest context of technological change. |

**Program Specific Outcomes**

| **PSO1** | An ability to analyze the common business functions to design and develop appropriate Information Technology solutions for social upliftments. |
| --- | --- |
| **PSO2** | Shall have expertise on the evolving technologies like Mobile Apps, CRM, ERP, Big Data, etc. |

**CRYPTOGRAPHY AND NETWORK SECURITY LAB**

**B.Tech. IV Year I Sem. L T P C**

**Course Code: CS703PC 0 0 2 1**

1. Write a C program that contains a string (char pointer) with a value ‘Hello world’.

The program should XOR each character in this string with 0 and displays the result.

2. Write a C program that contains a string (char pointer) with a value ‘Hello world’.

The program should AND or and XOR each character in this string with 127 and

display the result.

3. Write a Java program to perform encryption and decryption using the following

Algorithms

a. Ceaser cipher b. Substitution cipher c. Hill Cipher

4. Write a C/JAVA program to implement the DES algorithm logic.

5. Write a C/JAVA program to implement the Blowfish algorithm logic.

6. Write a C/JAVA program to implement the Rijndael algorithm logic.

7. Write the RC4 logic in Java Using Java cryptography; encrypt the text “Hello world” using Blowfish. Create your own key using Java key tool.

8. Write a Java program to implement RSA algorithm.

9. Implement the Diffie-Hellman Key Exchange mechanism using HTML and

JavaScript.

10. Calculate the message digest of a text using the SHA-1 algorithm in JAVA.

11. Calculate the message digest of a text using the MD5 algorithm in JAVA.

**COURSE OUTCOMES:**

Students can be able to

**CS326.CO1:** Student will be able to understand basic cryptographic algorithms, message and web authentication and security issues.

**CS326.CO** 2:Ability to identify information system requirements for both of them such as client and server.

**CS326.CO** 3: Ability to understand the current legal issues towards information security.

|  | **PO1** | **PO2** | **PO3** | **PO4** | **PO5** | **PO6** | **PO7** | **PO8** | **PO9** | **PO10** | **PO11** | **PO12** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **CO1** | **3** | **3** | **3** | **3** | **3** |  |  | **2** |  | **1** |  |  |
| **CO2** | **3** | **3** | **3** | **3** | **3** |  |  |  |  | **3** |  |  |
| **CO3** | **2** | **2** | **2** | **2** | **2** |  |  | **2** | **2** | **2** |  | **3** |

**LIST OF EXPERIMENTS**

| **EXP NO** | **EXPERIMENT NAME** |
| --- | --- |
| 1 | Write a C program that contains a string (char pointer) with a value ‘Hello world’. The program should XOR each character in this string with 0 and displays the result. |
| 2 | Write a C program that contains a string (char pointer) with a value ‘Hello world’. The program should AND or and XOR each character in this string with 127 and display the result. |
| 3 | Write a Java program to perform encryption and decryption using the following Algorithms:  a. Ceaser cipher  b. Substitution cipher  c. Hill Cipher |
| 4 | Write a C/JAVA program to implement the DES algorithm logic. |
| 5 | Write a C/JAVA program to implement the Blowfish algorithm logic. |
| 6 | Write a C/JAVA program to implement the Rijndael algorithm logic. |
| 7 | Write the RC4 logic in Java Using Java cryptography; encrypt the text “Hello world” using Blowfish. Create your own key using Java key tool. |
| 8 | Write a Java program to implement RSA algorithm. |
| 9 | Implement the Diffie-Hellman Key Exchange mechanism using HTML and JavaScript. |
| 10 | Calculate the message digest of a text using the SHA-1 algorithm in JAVA. |
| 11 | Calculate the message digest of a text using the MD5 algorithm in JAVA. |

**EXPERIMENT 1**

**1. Write a C program that contains a string (char pointer) with a value \Hello World’. The program should XOR each character in this string with 0 and displays the result.**

**PROGRAM:**

#include<stdlib.h>

main()

{

char str[]="Hello World";

char str1[11];

int i,len;

len=strlen(str);

for(i=0;i<len;i++)

{

str1[i]=str[i]^0;

printf("%c",str1[i]);

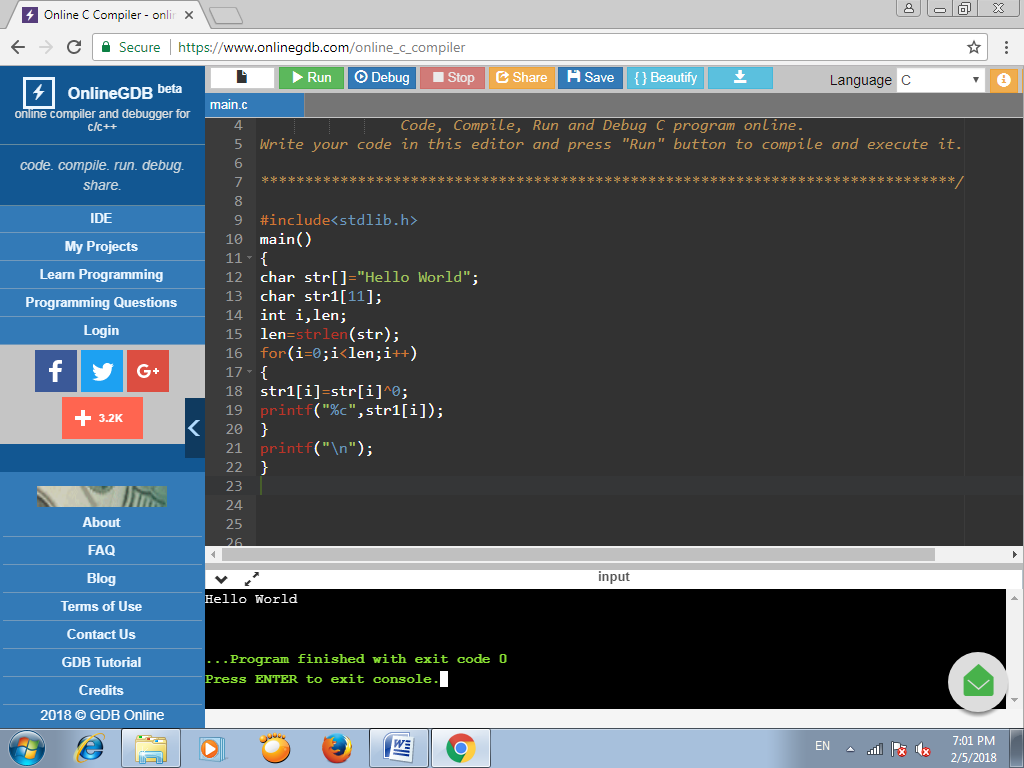
}

printf("\n");

}

**OUTPUT:**

Hello World



**EXPERIMENT 2**

**2. Write a C program that contains a string (char pointer) with a value ‘Hello world’.**

**The program should AND or and XOR each character in this string with 127 and display the result.**

**PROGRAM:**

#include <stdio.h>

#include<stdlib.h>

void main()

{

char str[]="Hello World";

char str1[11];

char str2[11];

int i,len;

len = strlen(str);

for(i=0;i<len;i++)

{

str1[i] = (str[i]&127) | (str[i]^127);

printf("%d",str1[i]);

}

printf("\n");

for(i=0;i<len;i++)

{

str2[i] = (str[i]&127) & (str[i]^127);

printf("%d",str2[i]);

}

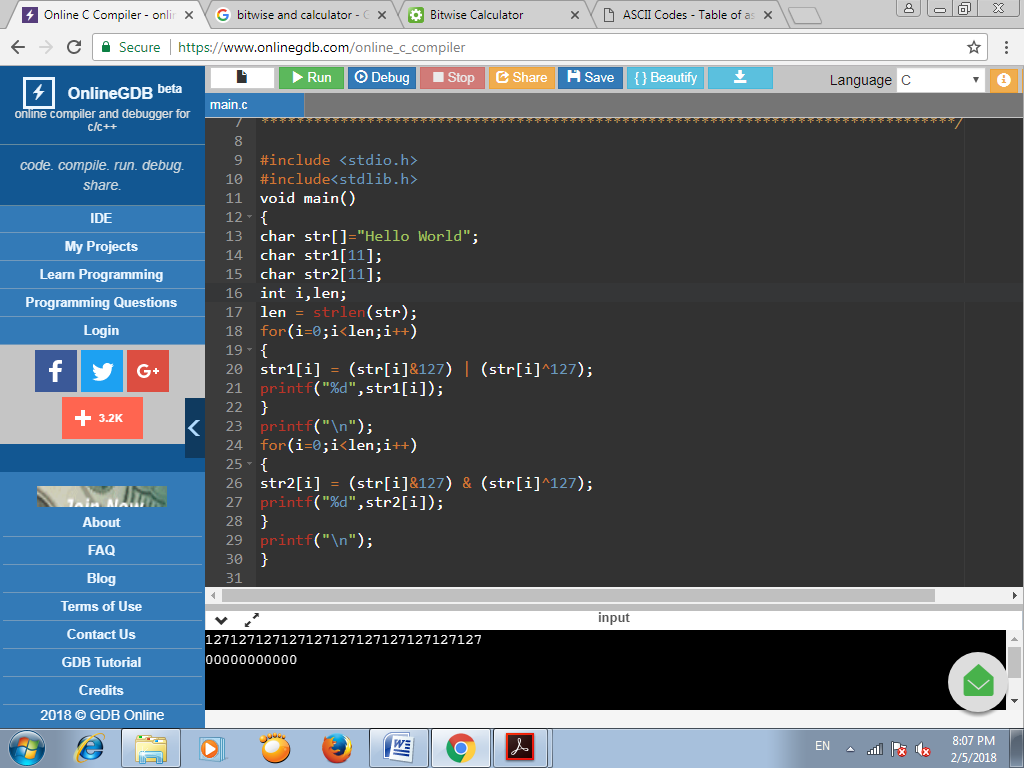
printf("\n");

}

**OUTPUT:**

127127127127127127127127127127127

00000000000



**EXPERIMENT 3**

**3. Write a Java program to perform encryption and decryption using the following**

**Algorithms**

**a. Ceaser cipher b. Substitution cipher c. Hill Cipher**

**PROGRAM:**

**a) Ceaser Cipher**

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

import java.util.Scanner;

public class CeaserCipher {

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

String str = "Hello World";

int key=5;

String encrypted = encrypt(str, key);

System.out.println("\nEncrypted String is: " +encrypted);

String decrypted = decrypt(encrypted, key);

System.out.println("\nDecrypted String is: " +decrypted);

System.out.println("\n");

}

public static String encrypt(String str, int key) {

String encrypted = "";

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

int c = str.charAt(i);

if (Character.isUpperCase(c)) {

c = c + (key % 26);

if (c > 'Z')

c = c - 26;

}

else if (Character.isLowerCase(c)) {

c = c + (key % 26);

if (c > 'z')

c = c - 26;

}

encrypted += (char) c;

}

return encrypted;

}

public static String decrypt(String str, int key) {

String decrypted = "";

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

int c = str.charAt(i);

if (Character.isUpperCase(c)) {

c = c - (key % 26);

if (c < 'A')

c = c + 26;

}

else if (Character.isLowerCase(c)) {

c = c - (key % 26);

if (c < 'a')

c = c + 26;

}

decrypted += (char) c;

}

return decrypted;

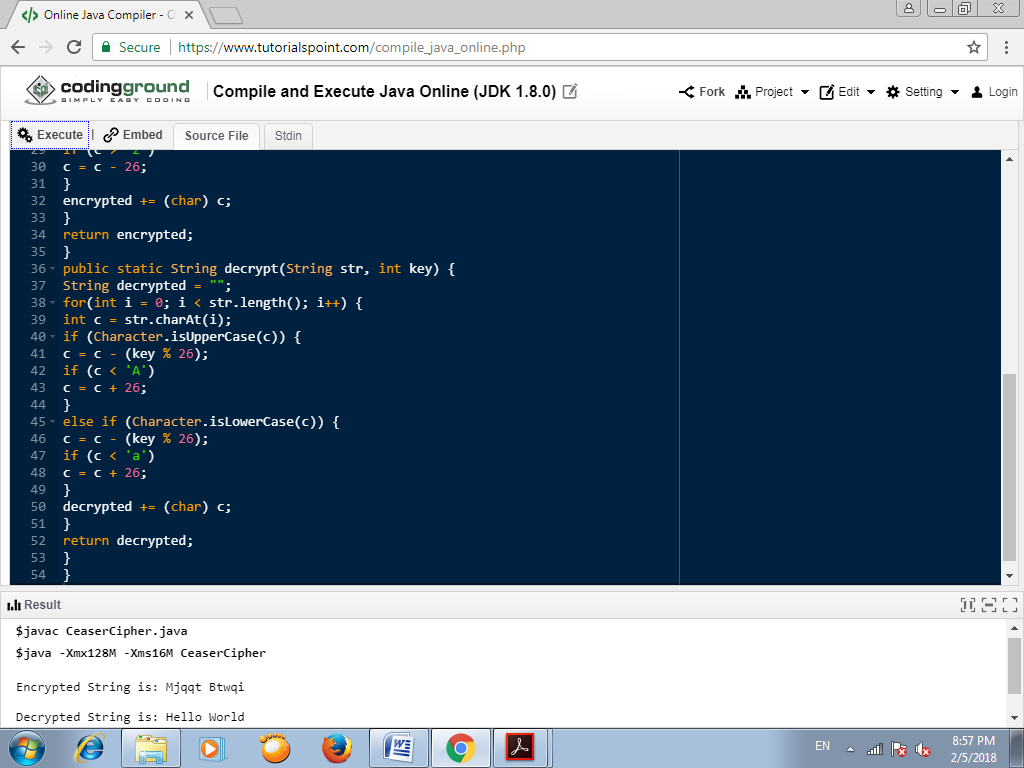
}

}

**OUTPUT:**

Encrypted String is: Mjqqt Btwqi

Decrypted String is: Hello World



**b) Substitution Cipher**

**PROGRAM:**

import java.io.\*;

import java.util.\*;

public class SubstitutionCipher {

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

String str = "hello world";

String a = "abcdefghijklmnopqrstuvwxyz ";

String b = "zyxwvutsrqponmlkjihgfedcba ";

String encrypt = "";

String decrypt = "";

char c;

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

{

c = str.charAt(i);

int j = a.indexOf(c);

encrypt = encrypt+b.charAt(j);

}

System.out.println("The encrypted data is: " +encrypt);

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

{

c = encrypt.charAt(i);

int j = a.indexOf(c);

decrypt = decrypt+b.charAt(j);

}

System.out.println("The decrypted data is: " +decrypt);

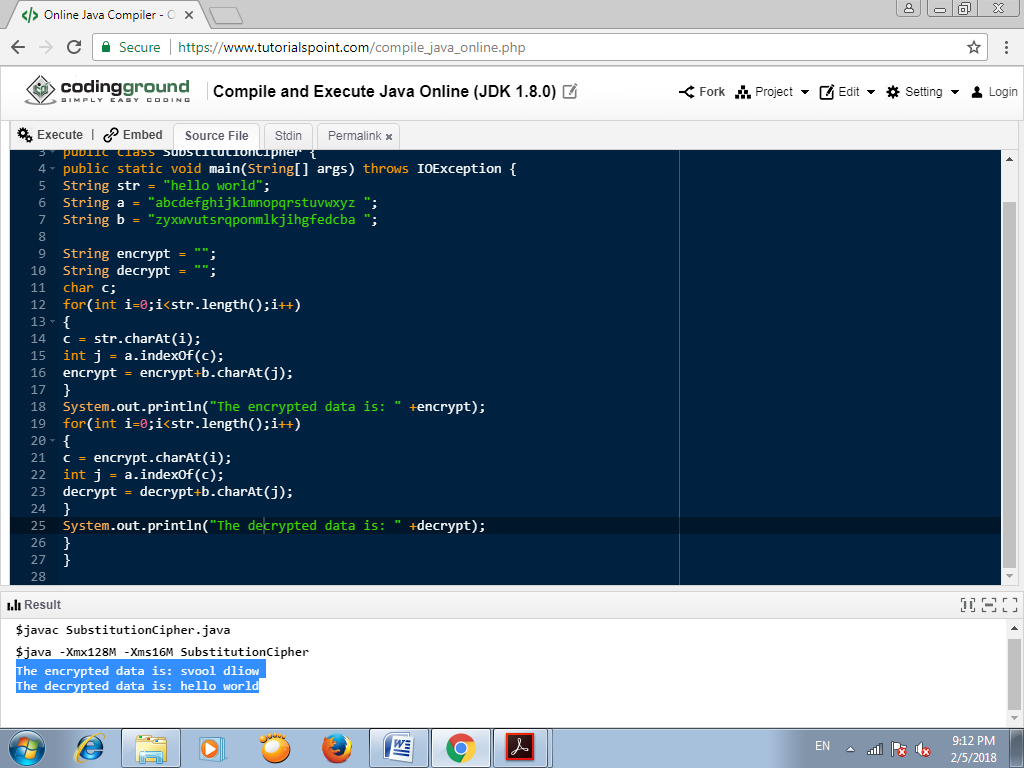
}

}

**OUTPUT:**

The encrypted data is: svool dliow

The decrypted data is: hello world



**c) Hill Cipher**

**PROGRAM:**

import java.io.\*;

import java.util.\*;

import java.io.\*;

public class HillCipher {

static int[][] decrypt = new int[3][1];

static int[][] b = new int[3][3];

static int[][] mes = new int[3][1];

static int[][] res = new int[3][1];

static int a[][] = {{1,2,3},{0,1,4},{5,6,0}};

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

getkeymes();

for(int i=0;i<3;i++)

for(int j=0;j<1;j++)

for(int k=0;k<3;k++) {

res[i][j]=res[i][j]+a[i][k]\*mes[k][j]; }

System.out.print("\nEncrypted string is : ");

for(int i=0;i<3;i++) {

System.out.print((char)(res[i][0]%26+97));

res[i][0]=res[i][0];

}

inverse();

for(int i=0;i<3;i++)

for(int j=0;j<1;j++)

for(int k=0;k<3;k++) {

decrypt[i][j] = decrypt[i][j]+b[i][k]\*res[k][j]; }

System.out.print("\nDecrypted string is : ");

for(int i=0;i<3;i++){

System.out.print((char)(decrypt[i][0]%26+97));

}

System.out.print("\n");

}

public static void getkeymes() throws IOException {

String msg = "cse";

for(int i=0;i<3;i++)

mes[i][0] = msg.charAt(i)-97;

}

public static void inverse() {

int p, q;

int[][] c = a;

for(int i=0;i<3;i++)

for(int j=0;j<3;j++) {

//a[i][j]=sc.nextint();

if(i==j)

b[i][j]=1;

else b[i][j]=0;

}

for(int k=0;k<3;k++) {

for(int i=0;i<3;i++) {

p = c[i][k];

q = c[k][k];

for(int j=0;j<3;j++) {

if(i!=k) {

c[i][j] = c[i][j]\*q-p\*c[k][j];

b[i][j] = b[i][j]\*q-p\*b[k][j];

} } } }

for(int i=0;i<3;i++)

for(int j=0;j<3;j++) {

b[i][j] = b[i][j]/c[i][i]; }

System.out.println("");

System.out.println("\nInverse Matrix is : ");

for(int i=0;i<3;i++) {

for(int j=0;j<3;j++)

System.out.print(b[i][j] + " ");

System.out.print("\n"); }

}

}

**OUTPUT:**

Encrypted string is : yio

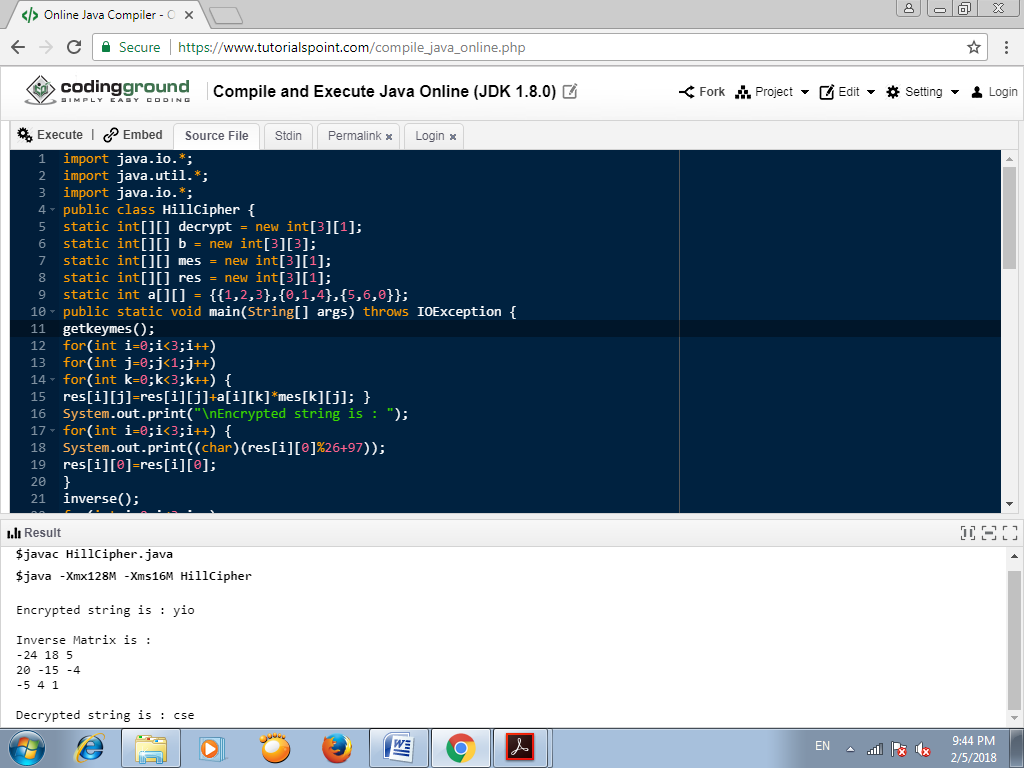
Inverse Matrix is :

-24 18 5

20 -15 -4

-5 4 1

Decrypted string is : cse



**EXPERIMENT 4**

**4. Write a C/JAVA program to implement the DES algorithm logic.**

**PROGRAM:**

import java.util.\*;

import java.io.BufferedReader;

import java.io.InputStreamReader;

import java.security.spec.KeySpec;

import javax.crypto.Cipher;

import javax.crypto.SecretKey;

import javax.crypto.SecretKeyFactory;

import javax.crypto.spec.DESedeKeySpec;

import sun.misc.BASE64Decoder;

import sun.misc.BASE64Encoder;

public class DES {

private static final String UNICODE\_FORMAT = "UTF8";

public static final String DESEDE\_ENCRYPTION\_SCHEME = "DESede";

private KeySpec myKeySpec;

private SecretKeyFactory mySecretKeyFactory;

private Cipher cipher;

byte[] keyAsBytes;

private String myEncryptionKey;

private String myEncryptionScheme;

SecretKey key;

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

public DES() throws Exception {

// TODO code application logic here

myEncryptionKey = "ThisIsSecretEncryptionKey";

myEncryptionScheme = DESEDE\_ENCRYPTION\_SCHEME;

keyAsBytes = myEncryptionKey.getBytes(UNICODE\_FORMAT);

myKeySpec = new DESedeKeySpec(keyAsBytes);

mySecretKeyFactory = SecretKeyFactory.getInstance(myEncryptionScheme);

cipher = Cipher.getInstance(myEncryptionScheme);

key = mySecretKeyFactory.generateSecret(myKeySpec);

}

public String encrypt(String unencryptedString) {

String encryptedString = null;

try {

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

byte[] plainText = unencryptedString.getBytes(UNICODE\_FORMAT);

byte[] encryptedText = cipher.doFinal(plainText);

BASE64Encoder base64encoder = new BASE64Encoder();

encryptedString = base64encoder.encode(encryptedText); }

catch (Exception e) {

e.printStackTrace(); }

return encryptedString; }

public String decrypt(String encryptedString) {

String decryptedText=null;

try {

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

BASE64Decoder base64decoder = new BASE64Decoder();

byte[] encryptedText = base64decoder.decodeBuffer(encryptedString);

byte[] plainText = cipher.doFinal(encryptedText);

decryptedText= bytes2String(plainText); }

catch (Exception e) {

e.printStackTrace(); }

return decryptedText; }

private static String bytes2String(byte[] bytes) {

StringBuffer stringBuffer = new StringBuffer();

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

stringBuffer.append((char) bytes[i]); }

return stringBuffer.toString(); }

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

DES myEncryptor= new DES();

String stringToEncrypt = "cse";

String encrypted = myEncryptor.encrypt(stringToEncrypt);

String decrypted = myEncryptor.decrypt(encrypted);

System.out.println("\nString To Encrypt: " +stringToEncrypt);

System.out.println("\nEncrypted Value : " +encrypted);

System.out.println("\nDecrypted Value : " +decrypted);

System.out.println("");

}

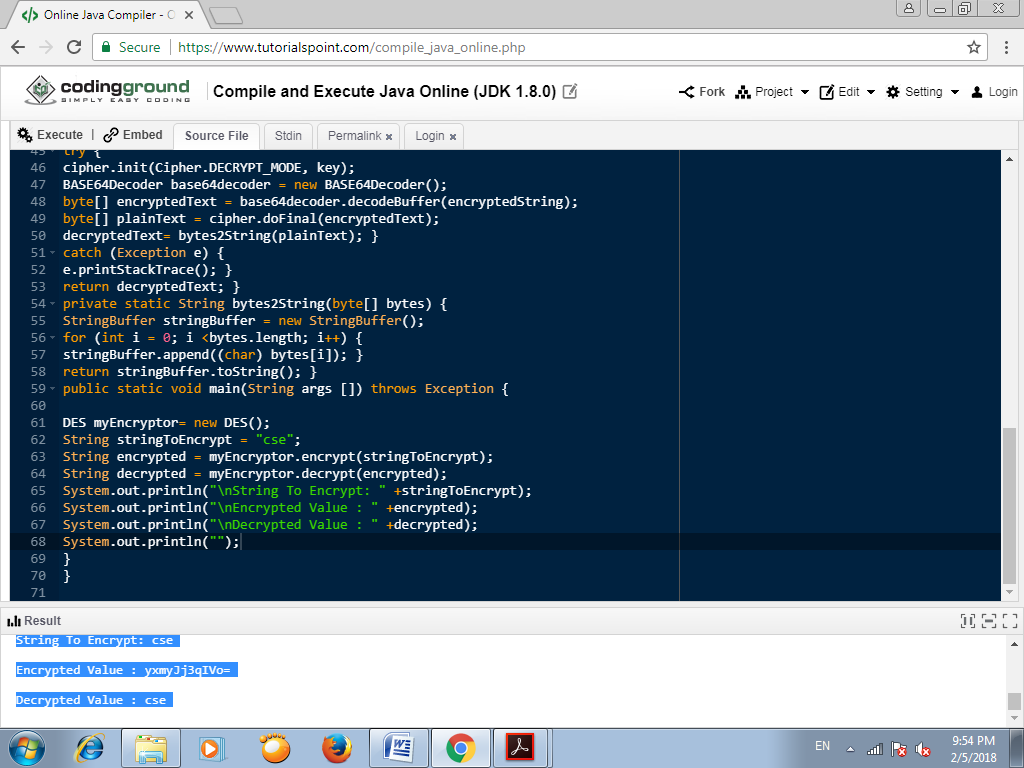
}

**OUTPUT:**

String To Encrypt: cse

Encrypted Value : yxmyJj3qIVo=

Decrypted Value : cse



**EXPERIMENT 5**

**5. Write a C/JAVA program to implement the Blowfish algorithm logic.**

**PROGRAM:**

import java.io.\*;

import java.io.FileInputStream;

import java.io.FileOutputStream;

import java.security.Key;

import javax.crypto.Cipher;

import javax.crypto.CipherOutputStream;

import javax.crypto.KeyGenerator;

import sun.misc.BASE64Encoder;

public class BlowFish {

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

// TODO code application logic here

KeyGenerator keyGenerator = KeyGenerator.getInstance("Blowfish");

keyGenerator.init(128);

Key secretKey = keyGenerator.generateKey();

Cipher cipherOut = Cipher.getInstance("Blowfish/CFB/NoPadding");

cipherOut.init(Cipher.ENCRYPT\_MODE, secretKey);

BASE64Encoder encoder = new BASE64Encoder();

byte iv[] = cipherOut.getIV();

if (iv != null) {

System.out.print("Initialization Vector of the Cipher: " + encoder.encode(iv)); }

FileInputStream fin = new FileInputStream("inputFile");

FileOutputStream fout = new FileOutputStream("outputFile.txt");

CipherOutputStream cout = new CipherOutputStream(fout, cipherOut);

int input = 0;

while ((input = fin.read()) != -1)

{

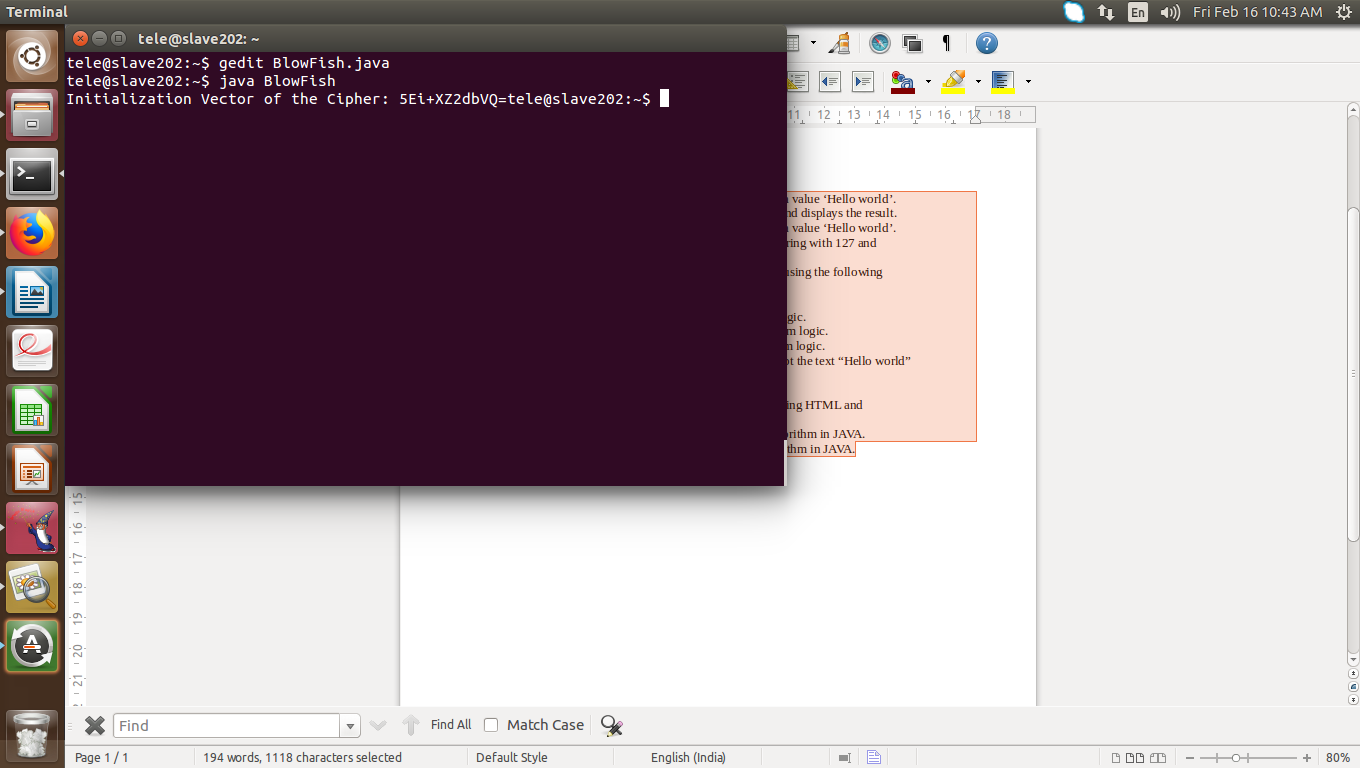
cout.write(input);

}

fin.close(); cout.close();

}

}

**OUTPUT:**

**EXPERIMENT 6**

**6. Write a C/JAVA program to implement the Rijndael algorithm logic.**

**PROGRAM:**

import java.security.\*;

import javax.crypto.\*;

import javax.crypto.spec.\*;

import java.io.\*;

public class AES {

public static String asHex (byte buf[]) {

StringBuffer strbuf = new StringBuffer(buf.length \* 2);

int i;

for (i = 0; i < buf.length; i++) {

if (((int) buf[i] & 0xff) < 0x10)

strbuf.append("0");

strbuf.append(Long.toString((int) buf[i] & 0xff, 16)); }

return strbuf.toString(); }

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

String message="AES still rocks!!";

// Get the KeyGenerator

KeyGenerator kgen = KeyGenerator.getInstance("AES");

kgen.init(128); // 192 and 256 bits may not be available

// Generate the secret key specs.

SecretKey skey = kgen.generateKey();

byte[] raw = skey.getEncoded();

SecretKeySpec skeySpec = new SecretKeySpec(raw, "AES");

// Instantiate the cipher

Cipher cipher = Cipher.getInstance("AES");

cipher.init(Cipher.ENCRYPT\_MODE, skeySpec);

byte[] encrypted = cipher.doFinal((args.length == 0 ? message : args[0]).getBytes());

System.out.println("encrypted string: " + asHex(encrypted));

cipher.init(Cipher.DECRYPT\_MODE, skeySpec);

byte[] original = cipher.doFinal(encrypted);

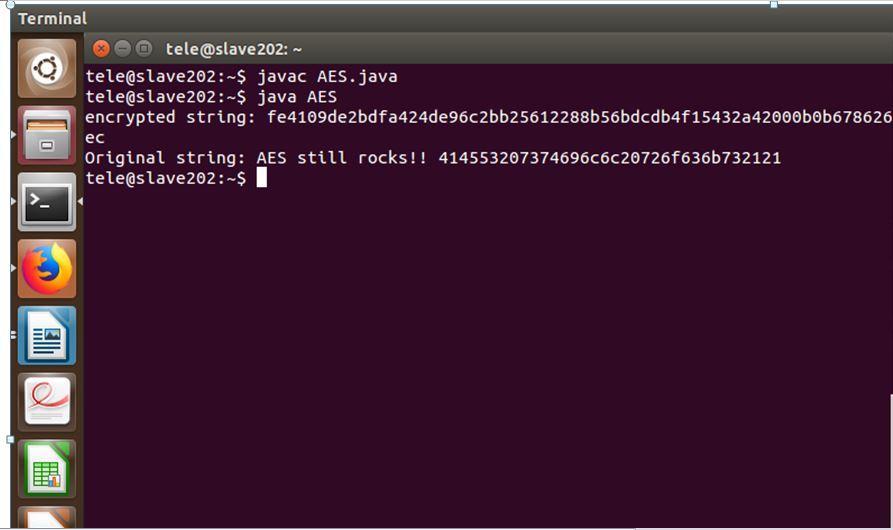
String originalString = new String(original);

System.out.println("Original string: " + originalString + " " + asHex(original)); }

//System.exit(0);

}

**OUTPUT:**



**EXPERIMENT 7**

**7. Write the RC4 logic in Java Using Java cryptography; encrypt the text “Hello world” using Blowfish. Create your own key using Java key tool.**

**AIM:** Using Java Cryptography, encrypt the text “Hello world” using BlowFish.

Create your own key using Java keytool.

**PROGRAM:**

**i**mport javax.crypto.Cipher;

import javax.crypto.KeyGenerator;

import javax.crypto.SecretKey;

import javax.swing.JOptionPane;

public class BlowFishCipher {

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

// create a key generator based upon the Blowfish cipher

KeyGenerator keygenerator = KeyGenerator.getInstance("Blowfish");

// create a key

SecretKey secretkey = keygenerator.generateKey();

// create a cipher based upon Blowfish

Cipher cipher = Cipher.getInstance("Blowfish");

// initialise cipher to with secret key

cipher.init(Cipher.ENCRYPT\_MODE, secretkey);

// get the text to encrypt

String inputText = JOptionPane.showInputDialog("Input your message: ");

// encrypt message

byte[] encrypted = cipher.doFinal(inputText.getBytes());

// re-initialise the cipher to be in decrypt mode

cipher.init(Cipher.DECRYPT\_MODE, secretkey);

// decrypt message

byte[] decrypted = cipher.doFinal(encrypted);

// and display the results

JOptionPane.showMessageDialog(JOptionPane.getRootFrame(),

"\nEncrypted text: " + new String(encrypted) + "\n" +

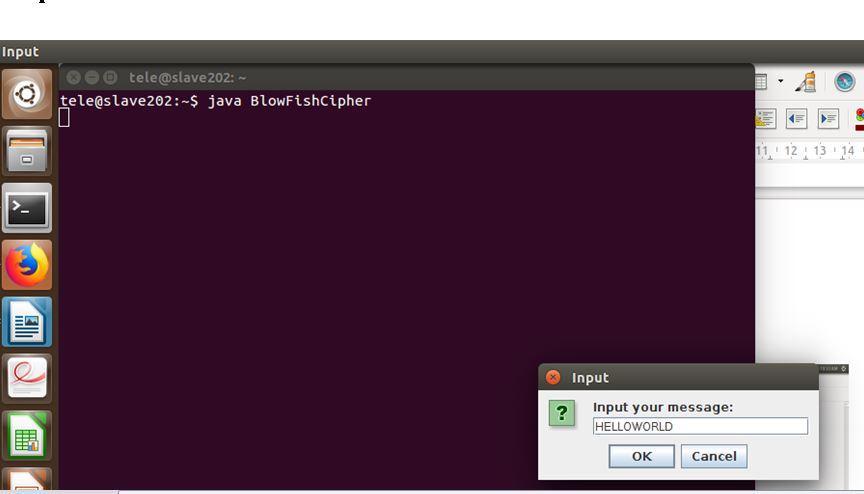
"\nDecrypted text: " + new String(decrypted));

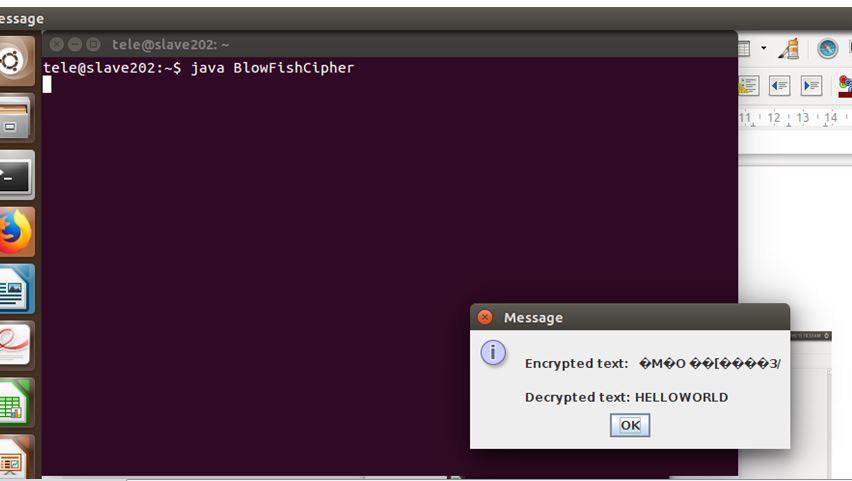
System.exit(0);

}

}

**OUTPUT:**





**EXPERIMENT 8**

**8) Write a Java program to implement 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();

}

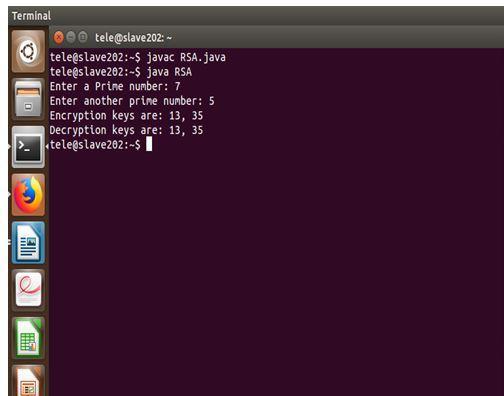
while(y <= 2 || intGCD != 1);

return e;

}

}

**OUTPUT:**

****

**EXPERIMENT 9**

**9) Implement the Diffie-Hellman Key Exchange mechanism using HTML and JavaScript.**

**PROGRAM:**

import java.math.BigInteger;

import java.security.KeyFactory;

import java.security.KeyPair;

import java.security.KeyPairGenerator;

import java.security.SecureRandom;

import javax.crypto.spec.DHParameterSpec;

import javax.crypto.spec.DHPublicKeySpec;

public class DiffeHellman {

public final static int pValue = 47;

public final static int gValue = 71;

public final static int XaValue = 9;

public final static int XbValue = 14;

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

// TODO code application logic here

BigInteger p = new BigInteger(Integer.toString(pValue));

BigInteger g = new BigInteger(Integer.toString(gValue));

BigInteger Xa = new BigInteger(Integer.toString(XaValue));

BigInteger Xb = new BigInteger(Integer.toString(XbValue));

createKey();

int bitLength = 512; // 512 bits

SecureRandom rnd = new SecureRandom();

p = BigInteger.probablePrime(bitLength, rnd);

g = BigInteger.probablePrime(bitLength, rnd);

createSpecificKey(p, g);

}

public static void createKey() throws Exception {

KeyPairGenerator kpg = KeyPairGenerator.getInstance("DiffieHellman");

kpg.initialize(512);

KeyPair kp = kpg.generateKeyPair();

KeyFactory kfactory = KeyFactory.getInstance("DiffieHellman");

DHPublicKeySpec kspec = (DHPublicKeySpec) kfactory.getKeySpec(kp.getPublic(),

DHPublicKeySpec.class);

System.out.println("Public key is: " +kspec);

}

public static void createSpecificKey(BigInteger p, BigInteger g) throws Exception {

KeyPairGenerator kpg = KeyPairGenerator.getInstance("DiffieHellman");

DHParameterSpec param = new DHParameterSpec(p, g);

kpg.initialize(param);

KeyPair kp = kpg.generateKeyPair();

KeyFactory kfactory = KeyFactory.getInstance("DiffieHellman");

DHPublicKeySpec kspec = (DHPublicKeySpec) kfactory.getKeySpec(kp.getPublic(),

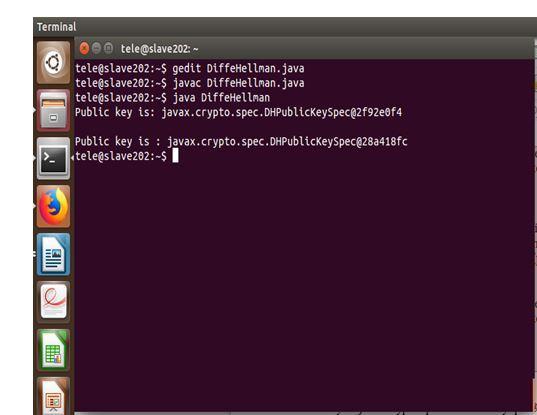
DHPublicKeySpec.class);

System.out.println("\nPublic key is : " +kspec);

}

}

**OUTPUT:**



**EXPERIMENT 10**

**10. Calculate the message digest of a text using the SHA-1 algorithm in JAVA.**

**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(""); }

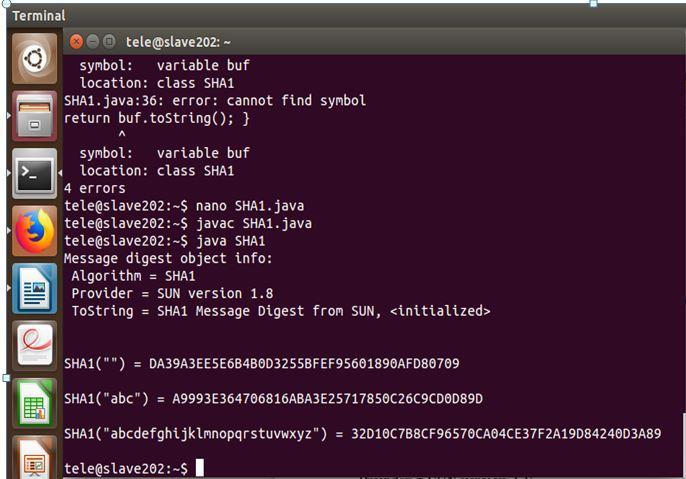
catch (Exception e) {

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

}

}

**OUTPUT:**



**EXPERIMENT 11**

**11. Calculate the message digest of a text using the MD5 algorithm in JAVA.**

**PROGRAM:**

**i**mport java.security.\*;

public class MD5

{

public static void main(String[] a)

{

// TODO code application logic here

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]); }

}

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:**

