CRYPTOGRAPHY AND NETWORK SECURITY LAB

B.Tech. III Year II Sem.

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.

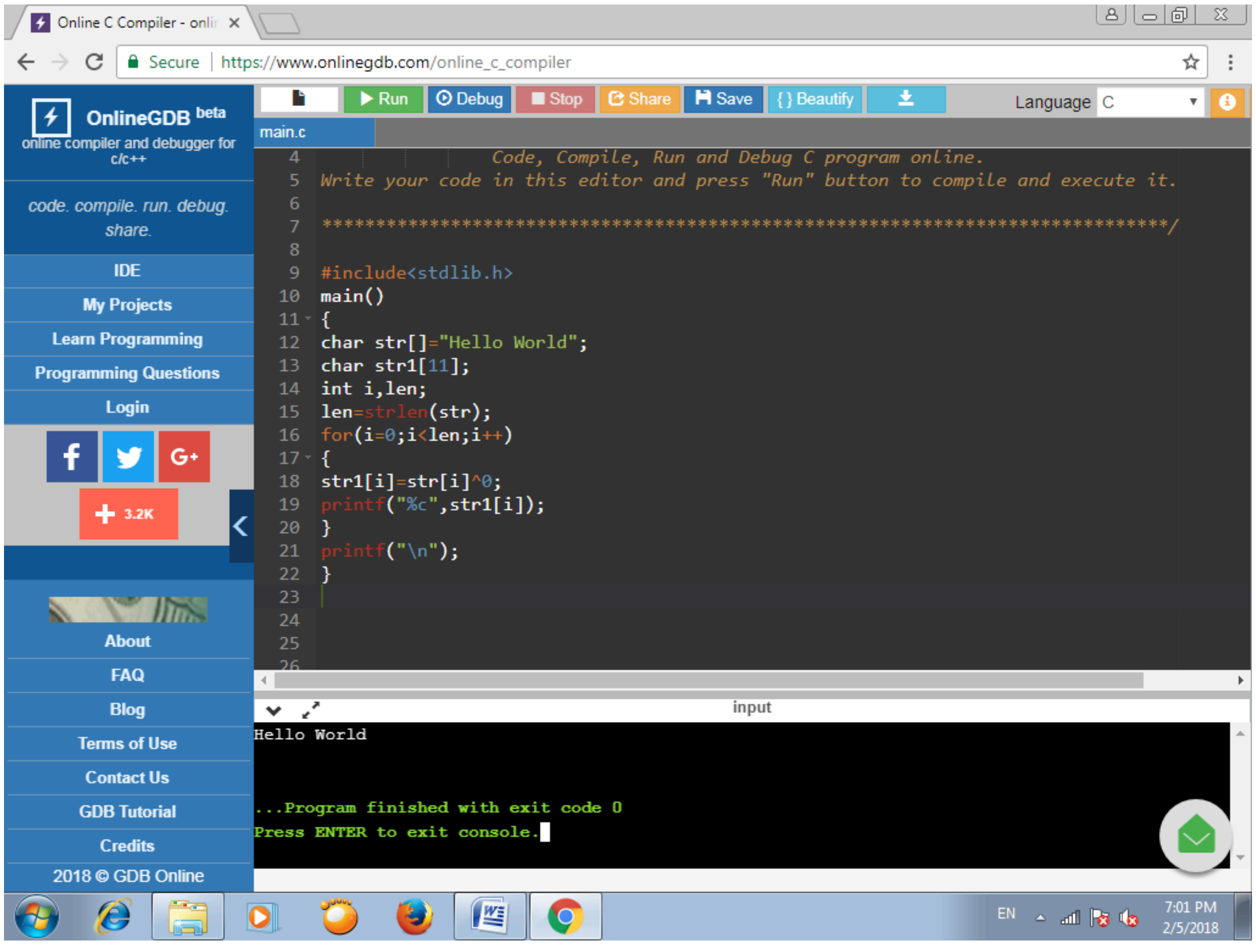
11. Calculate the message *digest of a text* using the MD5 algorithm in JAVA.

CRYPTOGRAPHY & NETWORK SECURITY LAB

1. **XOR a string with a Zero**

AIM: 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 display 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
```



```
                Code, Compile, Run and Debug C program online.
Write your code in this editor and press "Run" button to compile and execute it.

*******************************************************************************/

#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");
}
```

```
Hello World


...Program finished with exit code 0
Press ENTER to exit console.
```

2. **XOR a string with a 127**

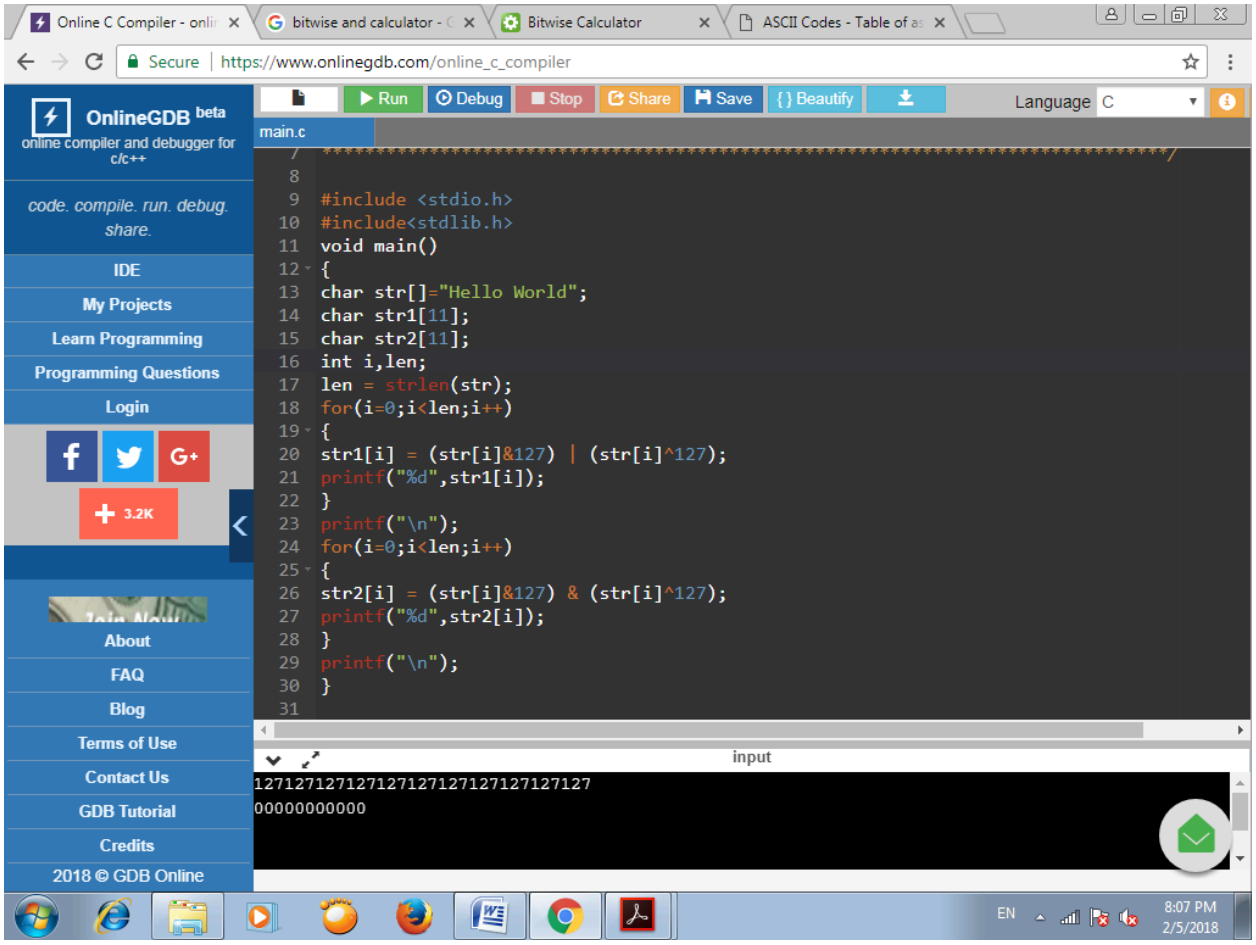
AIM: Write a C program that contains a string (char pointer) with a value \Hello World’. The program should AND or and XOR each character in this string with 127 and display the result.

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
```



```c
#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");
}
```

```
127127127127127127127127127127127
00000000000
```

3. **Encryption & Decryption using Cipher Algorithms**

AIM: Write a Java program to perform encryption and decryption using the following algorithms:

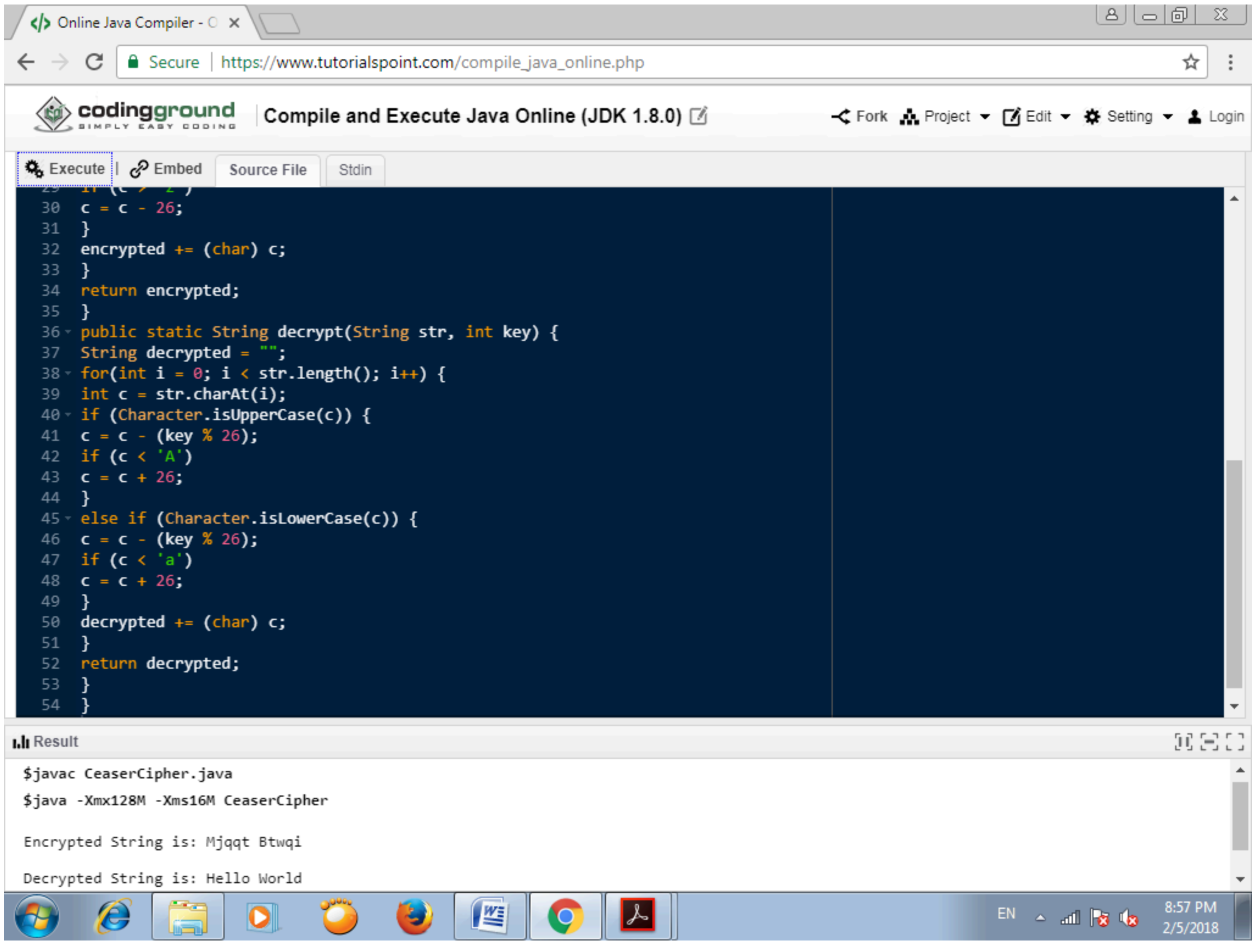
a) Ceaser Cipher
b) Substitution Cipher
c) Hill Cipher

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
```


Online Java Compiler
Secure https://www.tutorialspoint.com/compile_java_online.php
codingground SIMPLY EASY CODING
Compile and Execute Java Online (JDK 1.8.0)
Fork
Project
Edit
Setting
Login
Execute
Embed
Source File
Stdin
Result

```
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;
}
}
```

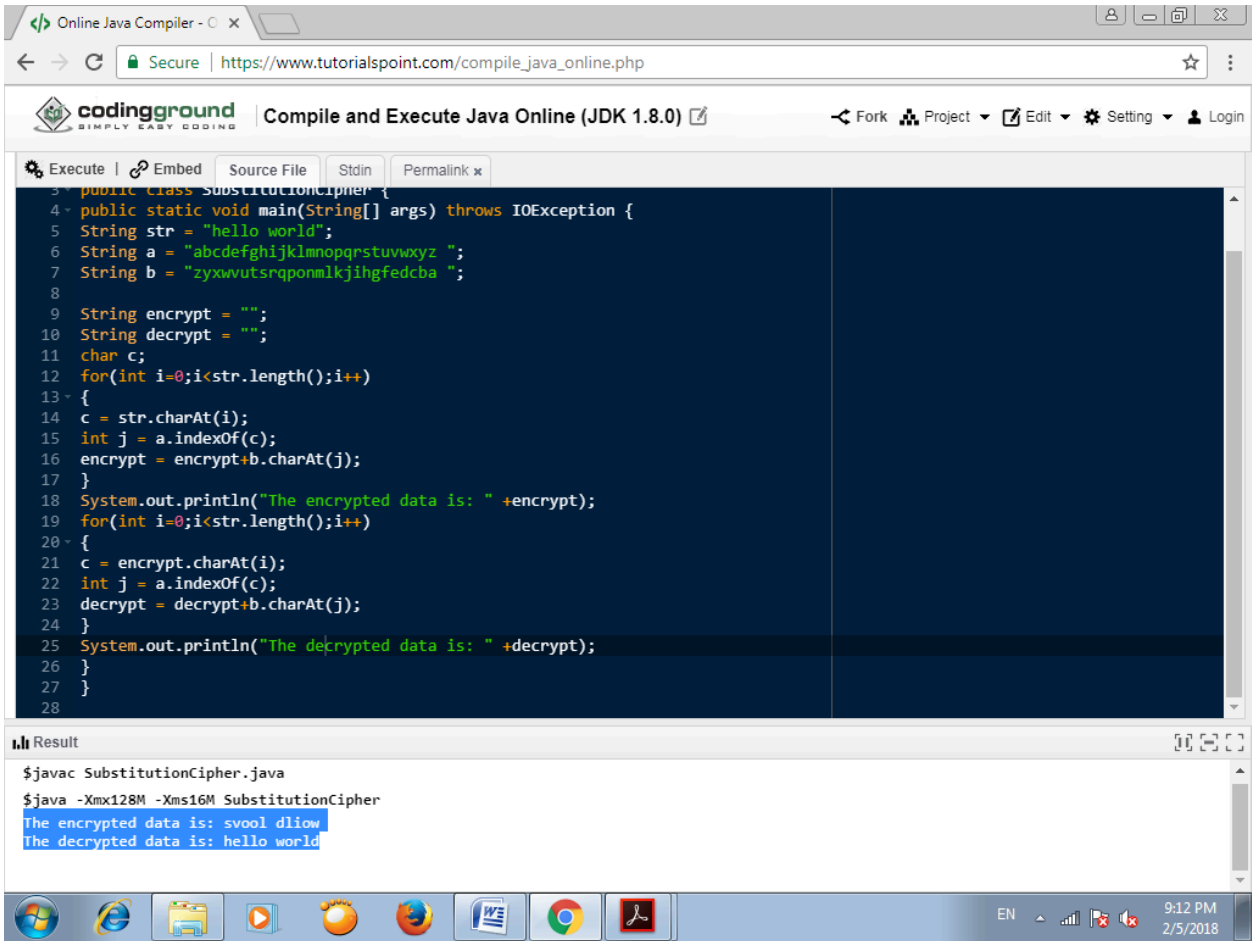
```
$javac CeaserCipher.java
$java -Xmx128M -Xms16M CeaserCipher

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
```



```
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);
}
}
```

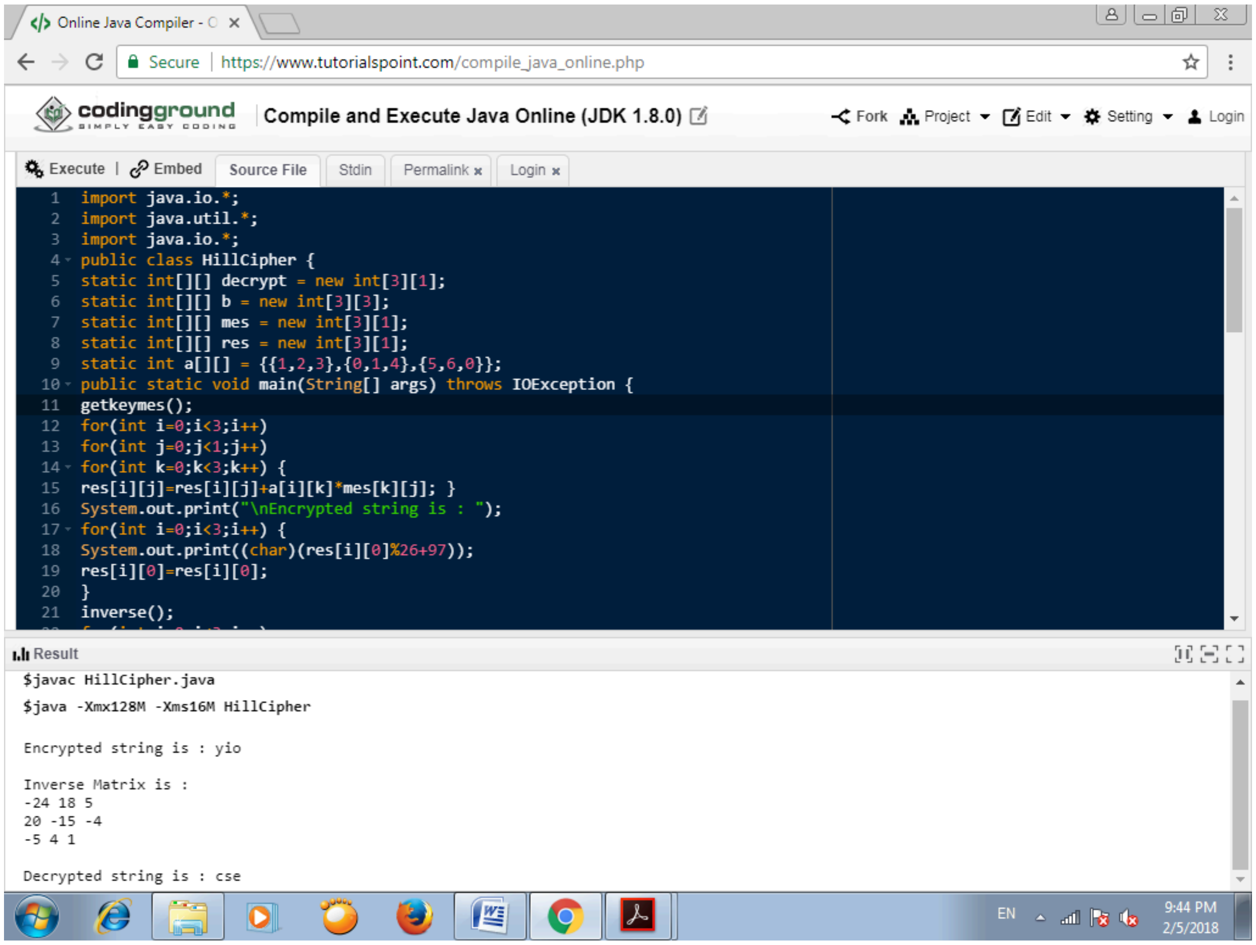
```
$javac SubstitutionCipher.java

$java -Xmx128M -Xms16M SubstitutionCipher
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
```



```java
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();
```

```
$javac HillCipher.java
$java -Xmx128M -Xms16M HillCipher

Encrypted string is : yio

Inverse Matrix is :
-24 18 5
20 -15 -4
-5 4 1

Decrypted string is : cse
```

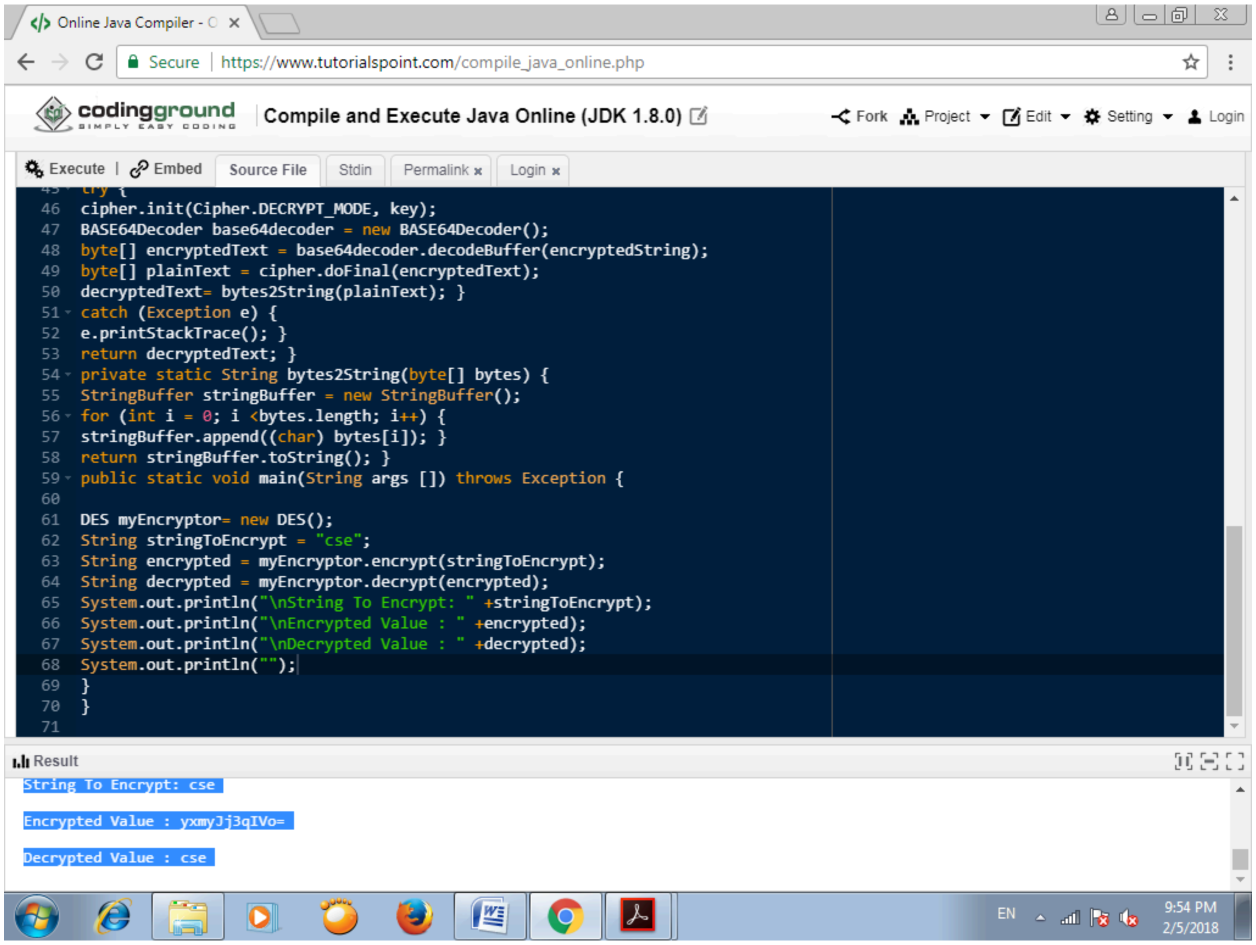
4. **Java program for DES algorithm logic**

AIM: Write a 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
```



Online Java Compiler - C
Secure | https://www.tutorialspoint.com/compile_java_online.php
codingground SIMPLY EASY CODING
Compile and Execute Java Online (JDK 1.8.0)
Fork
Project
Edit
Setting
Login
Execute
Embed
Source File
Stdin
Permalink
Login
Result
EN
9:54 PM
2/5/2018

```java
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("");
}
}
```

```
String To Encrypt: cse

Encrypted Value : yxmyJj3qIVo=

Decrypted Value : cse
```

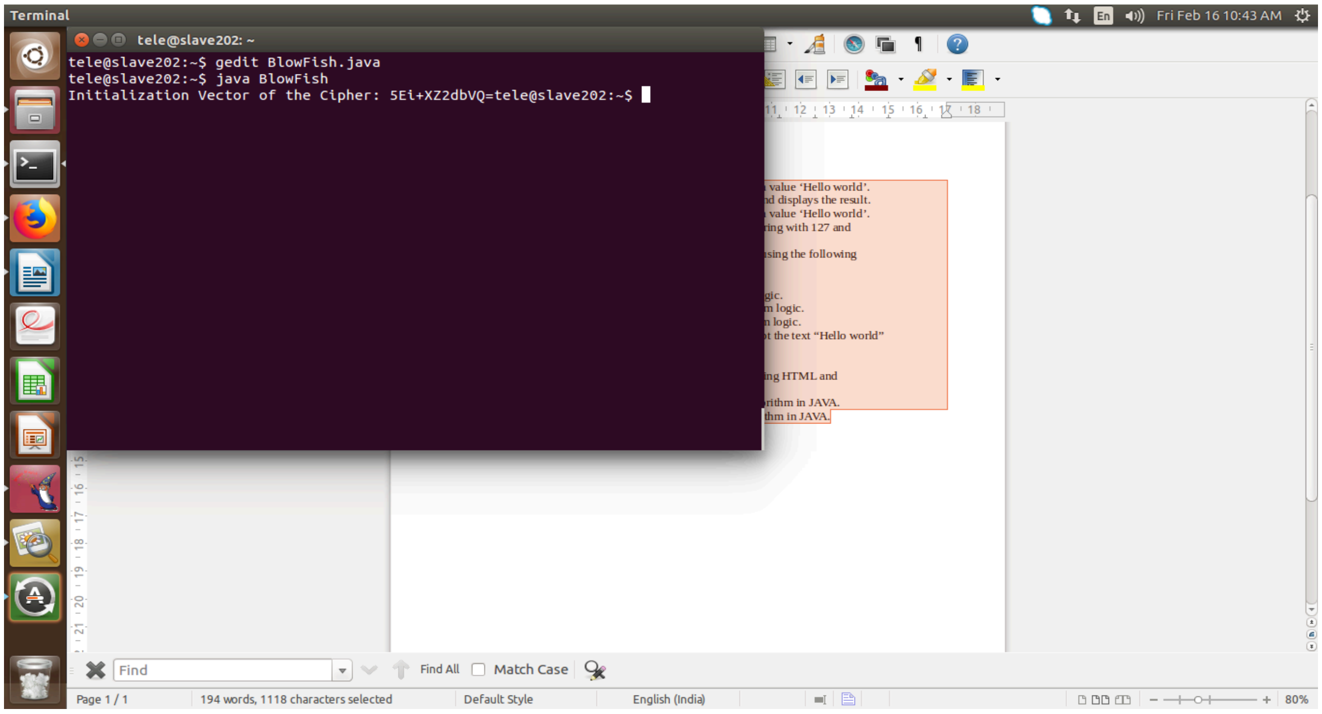
5. **JAVA program to implement the Blowfish algorithm logic**

AIM: 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:



```
tele@slave202:~$ gedit BlowFish.java
tele@slave202:~$ java BlowFish
Initialization Vector of the Cipher: 5Ei+XZ2dbVQ=tele@slave202:~$
```

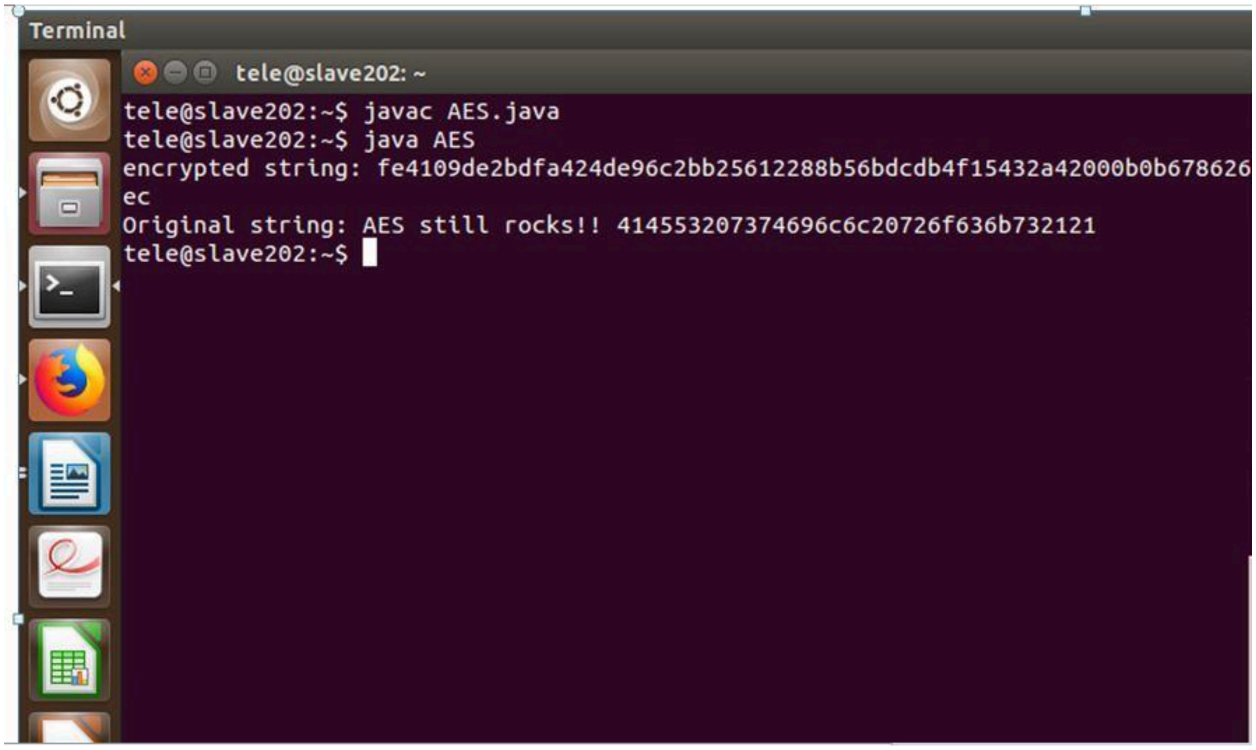
6. Program to implement Rijndael algorithm logic

AIM: 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:



```
tele@slave202:~$ javac AES.java
tele@slave202:~$ java AES
encrypted string: fe4109de2bdfa424de96c2bb25612288b56bdcdb4f15432a42000b0b678626
ec
Original string: AES still rocks!! 414553207374696c6c20726f636b732121
tele@slave202:~$
```

7.Encrypt a string using BlowFish algorithm

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

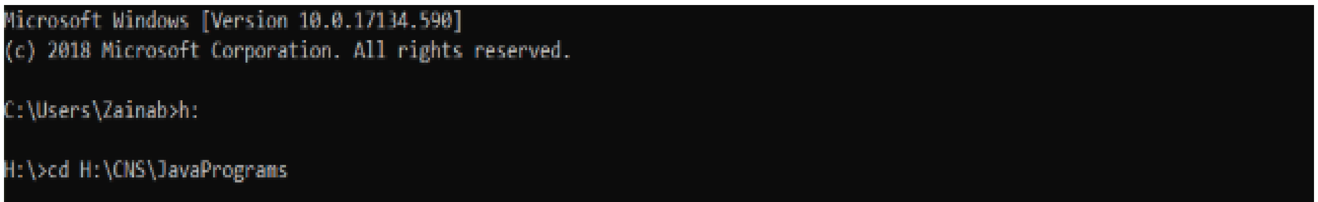
Create your own key using Java keytool.

PROGRAM:

```
import 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 keygenerator based upon the
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 secretkey
 cipher.init(Cipher.ENCRYPT_MODE,secretkey);
//get the text to encrypt
 String inputText = "Hello world";
//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
 System.out.println("Original String: " + inputText);
 System.out.println("Encrypted: " + new String(encrypted));
 System.out.println("Decrypted: " + new String(decrypted));
}
}
```

Output:



```
Microsoft Windows [Version 10.0.17134.590]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Zainab>h:

H:\>cd H:\CNS\JavaPrograms
```

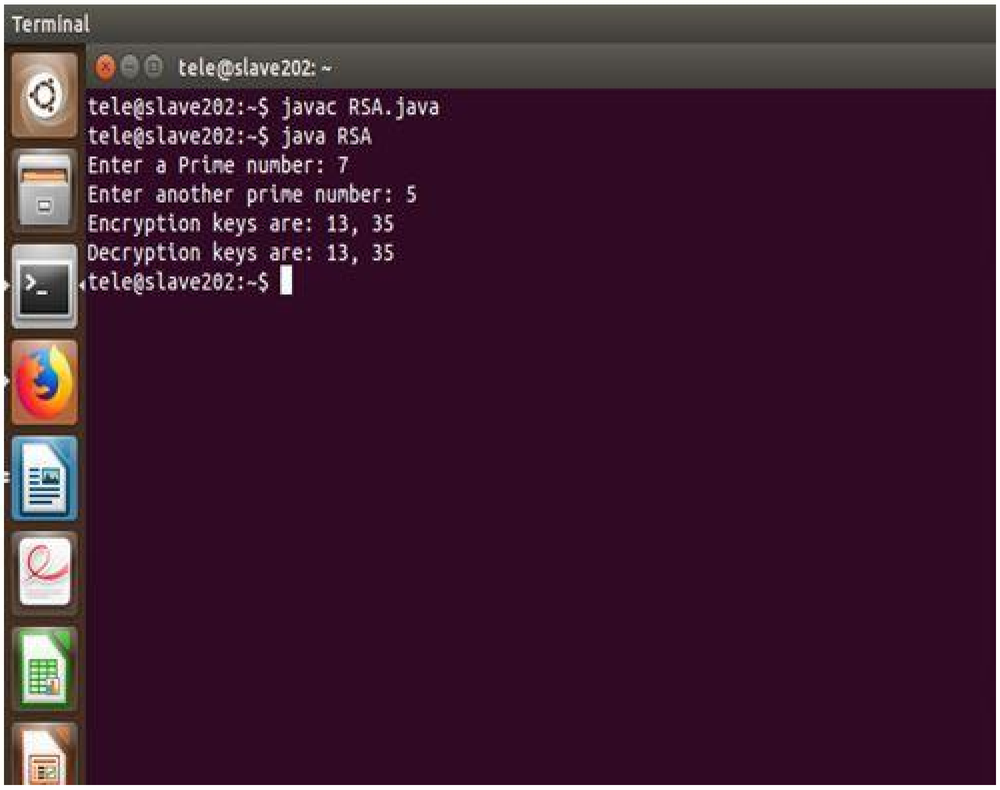
8) **Program to implement RSA Algorithm**

AIM: Write a Java program to implement RSA Algoithm.

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:



```
tele@slave202:~$ javac RSA.java
tele@slave202:~$ java RSA
Enter a Prime number: 7
Enter another prime number: 5
Encryption keys are: 13, 35
Decryption keys are: 13, 35
tele@slave202:~$
```

9) Diffie-Hellman

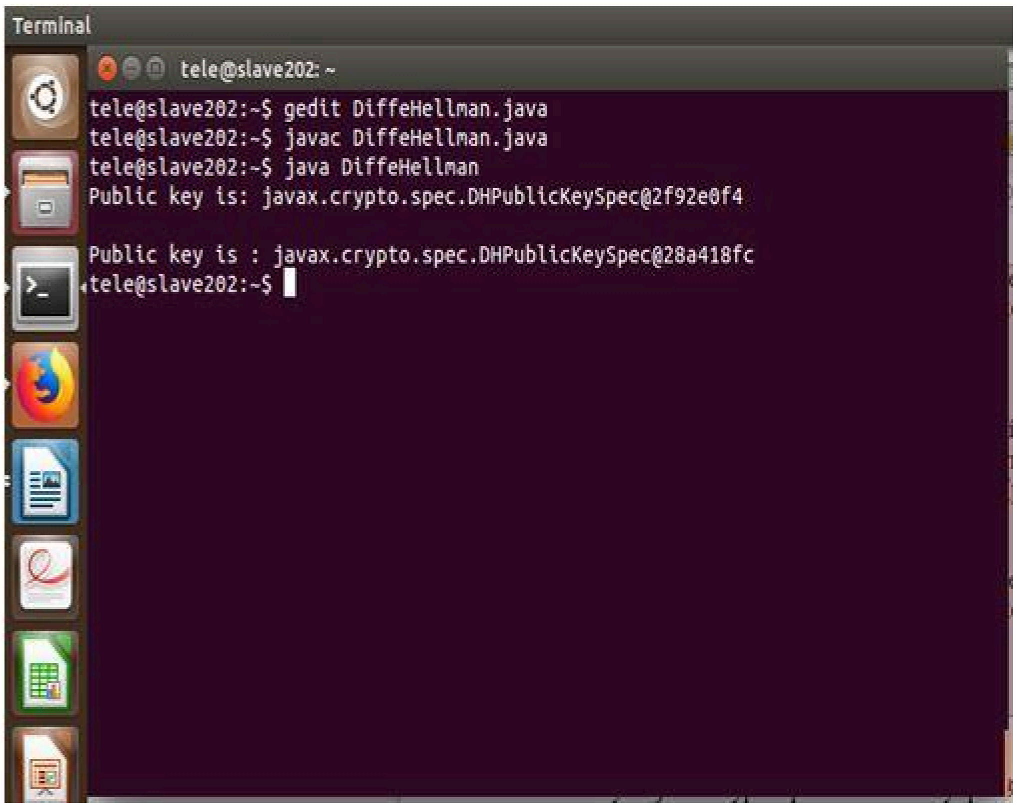
AIM: Implement the Diffie-Hellman Key Exchange mechanism using HTML and JavaScript. Consider the end user as one of the parties (Alice) and the JavaScript application as other party (bob).

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:



```
tele@slave202:~$ gedit DiffeHellman.java
tele@slave202:~$ javac DiffeHellman.java
tele@slave202:~$ java DiffeHellman
Public key is: javax.crypto.spec.DHPublicKeySpec@2f92e0f4

Public key is : javax.crypto.spec.DHPublicKeySpec@28a418fc
tele@slave202:~$
```

10. SHA-1

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

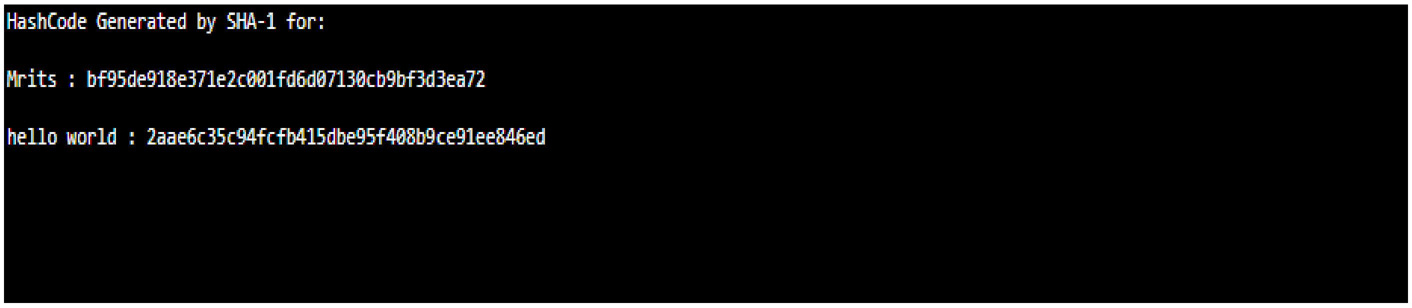
PROGRAM:

```
import java.math.BigInteger;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
 public class GFG {
   public static String encryptThisString(String input)
   {
       try {
           // getInstance() method is called with algorithm SHA-1
           MessageDigest md = MessageDigest.getInstance("SHA-1");

           // digest() method is called
           // to calculate message digest of the input string
           // returned as array of byte
```

```
            byte[] messageDigest = md.digest(input.getBytes());

            // Convert byte array into signum representation
            BigInteger no = new BigInteger(1, messageDigest);

            // Convert message digest into hex value
            String hashtext = no.toString(16);

            // Add preceding 0s to make it 32 bit
            while (hashtext.length() < 32) {
                hashtext = "0" + hashtext;
            }

            // return the HashText
            return hashtext;
        }

        // For specifying wrong message digest algorithms
        catch (NoSuchAlgorithmException e) {
            throw new RuntimeException(e);
        }
    }

    // Driver code
    public static void main(String args[]) throws
```

```
                                NoSuchAlgorithmException
    {


        System.out.println("HashCode Generated by SHA-1 for: ");


        String s1 = "Mrits";
        System.out.println("\n" + s1 + " : " + encryptThisString(s1));


        String s2 = "hello world";
        System.out.println("\n" + s2 + " : " + encryptThisString(s2));
    }
}
```

Output:



```
HashCode Generated by SHA-1 for:

Mrits : bf95de918e371e2c001fd6d07130cb9bf3d3ea72

hello world : 2aae6c35c94fcfb415dbe95f408b9ce91ee846ed
```

11. **MD5**

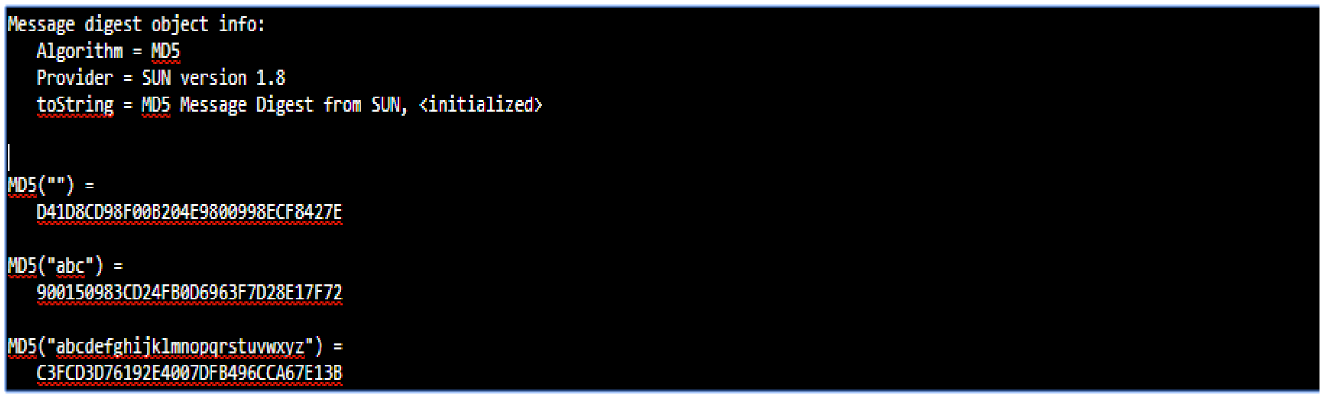
AIM: Calculate the message digest of a text using the SHA-1 algorithm in JAVA.

PROGRAM:

```
import java.security.*;
class JceSha1Test {
  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+"\") =");
      System.out.println("   "+bytesToHex(output));

      input = "abc";
      md.update(input.getBytes());
         output = md.digest();
      System.out.println();
      System.out.println("MD5(\""+input+"\") =");
      System.out.println("   "+bytesToHex(output));

      input = "abcdefghijklmnopqrstuvwxyz";
      md.update(input.getBytes());
         output = md.digest();
      System.out.println();
      System.out.println("MD5(\""+input+"\") =");
      System.out.println("   "+bytesToHex(output));

   } 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:



```
Message digest object info:
   Algorithm = MD5
   Provider = SUN version 1.8
   toString = MD5 Message Digest from SUN, <initialized>

MD5("") =
   D41D8CD98F00B204E9800998ECF8427E

MD5("abc") =
   900150983CD24FB0D6963F7D28E17F72

MD5("abcdefghijklmnopqrstuvwxyz") =
   C3FCD3D76192E4007DFB496CCA67E13B
```