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

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 key generator based upon the Blowfish

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

// encrypt

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

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

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

// decrypt

messagebyte[] 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:

Input your message: Hello world

Encrypted text: 3ooo&&(\*&\*4r4

Decrypted text: Hello world

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

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!!";

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

Kgen.init(128);

SecretKey sKey=Kgen.generateKey();

byte[]raw=sKey.getEncoded();

SecretKey sKeySpec=new SecretKeySpec(raw,"AES");

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);

StringoriginalString=new String(original);

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

}

}

OUTPUT:

Encrypted String:ae9d62f7416437d75a02cc66686d8bea

Original String:Rama