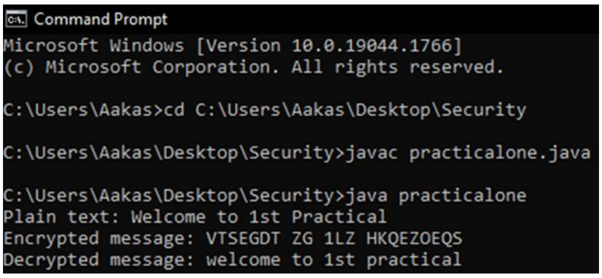
1. Implementation of the Mono-alphabetic Substitution Cipher using Frequency analysis method.

```
import java.io.*;
class practicalone {
    public static char normalChar[]
        = { 'a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i',
            'j', 'k', 'l', 'm', 'n', 'o', 'p', 'q', 'r',
            's', 't', 'u', 'v', 'w', 'x', 'y', 'z' };

    public static char codedChar[]
        = { 'Q', 'W', 'E', 'R', 'T', 'Y', 'U', 'I', 'O',
            'P', 'A', 'S', 'D', 'F', 'G', 'H', 'J', 'K',
            'L', 'Z', 'X', 'C', 'V', 'B', 'N', 'M' };
// Function which returns encrypted string
    public static String stringEncryption(String s)
    {
// initializing an empty String
        String encryptedString = "";
// comparing each character of the string and
// encoding each character using the indices
        for (int i = 0; i < s.length(); i++) {
            for (int j = 0; j < 26; j++) {
// comparing the character and
// adding the corresponding char
// to the encryptedString
                if (s.charAt(i) == normalChar[j])
                {
                    encryptedString += codedChar[j];
                    break;
                }
// if there are any special characters
// add them directly to the string
                if (s.charAt(i) < 'a' || s.charAt(i) > 'z')
                {
                    encryptedString += s.charAt(i);
                    break;
                }
            }
        }
// return encryptedString
        return encryptedString;
    }
// Function which returns descryptedString
    public static String stringDecryption(String s)
    {
// Initializing the string
        String decryptedString = "";
// Run the for loop for total string
        for (int i = 0; i < s.length(); i++)
        {
            for (int j = 0; j < 26; j++) {
// compare each characters and decode them
// using indices
                if (s.charAt(i) == codedChar[j])
                {
                    decryptedString += normalChar[j];
                    break;
                }
// Add the special characters directly to
// the String
                if (s.charAt(i) < 'A' || s.charAt(i) > 'Z')
                {
                    decryptedString += s.charAt(i);
                    break;
                }
            }
        }
// return the decryptedString
        return decryptedString;
    }
    public static void main(String args[])
    {
        String str = "Welcome to 1st Practical";
// print plain text
        System.out.println("Plain text: " + str);
// Changing whole string to lower case
// function call to stringEncryption and storing in
// encryptedString
        String encryptedString = stringEncryption(str.toLowerCase());
// printing encryptedString
        System.out.println("Encrypted message: "
                                + encryptedString);
// function call to stringDecryption and printing
// the decryptedString
        System.out.println("Decrypted message: "
            + stringDecryption(encryptedString));
    }
}
```

Output:

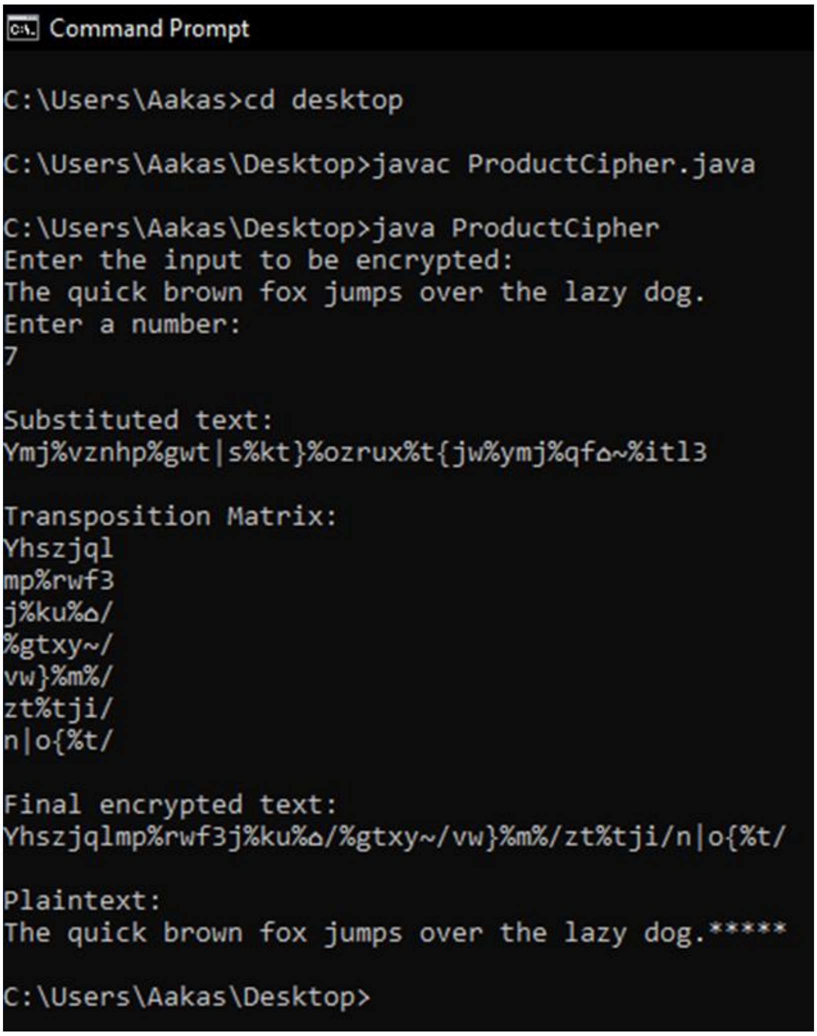


```
Microsoft Windows [Version 10.0.19044.1766]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Aakas>cd C:\Users\Aakas\Desktop\Security

C:\Users\Aakas\Desktop\Security>javac practicalone.java

C:\Users\Aakas\Desktop\Security>java practicalone
Plain text: Welcome to 1st Practical
Encrypted message: VTSEGDT ZG 1LZ HKQEZOEQS
Decrypted message: welcome to 1st practical
```

2. Design and implement a product cipher using Substitution ciphers.

```
import java.util.*;

class ProductCipher {
public static void main(String args[]) {
System.out.println("Enter the input to be encrypted:");
String substitutionInput = new Scanner(System.in).nextLine();
System.out.println("Enter a number:");
int n = new Scanner(System.in).nextInt();

// Substitution encryption
StringBuffer substitutionOutput = new StringBuffer();
for(int i=0 ; i<substitutionInput.length() ; i++) {
char c = substitutionInput.charAt(i);
substitutionOutput.append((char) (c+5));
}
System.out.println("\nSubstituted text:");
System.out.println(substitutionOutput);

// Transposition encryption
String transpositionInput = substitutionOutput.toString();
int modulus;
if((modulus = transpositionInput.length()%n) != 0) {
modulus = n-modulus;
// 'modulus' is now the number of blanks/padding (X) to be appended
for(;modulus!=0 ;modulus--) {
transpositionInput += "/";
}
}
StringBuffer transpositionOutput = new StringBuffer();
System.out.println("\nTransposition Matrix:");
for(int i=0 ; i<n ; i++) {
for(int j=0 ; j<transpositionInput.length()/n ; j++) {
char c = transpositionInput.charAt(i+(j*n));
System.out.print(c);
transpositionOutput.append(c);
}
System.out.println();
}
System.out.println("\nFinal encrypted text:");
System.out.println(transpositionOutput);

// Transposition decryption
n = transpositionOutput.length()/n;
StringBuffer transpositionPlaintext = new StringBuffer();
for(int i=0 ; i<n ; i++) {
for(int j=0 ; j<transpositionOutput.length()/n ; j++) {
char c = transpositionOutput.charAt(i+(j*n));
transpositionPlaintext.append(c);
}
}

// Substitution decryption
StringBuffer plaintext = new StringBuffer();
for(int i=0 ; i<transpositionPlaintext.length() ; i++) {
char c = transpositionPlaintext.charAt(i);
plaintext.append((char) (c-5));
}

System.out.println("\nPlaintext:");
System.out.println(plaintext);
}
}
```

Output:

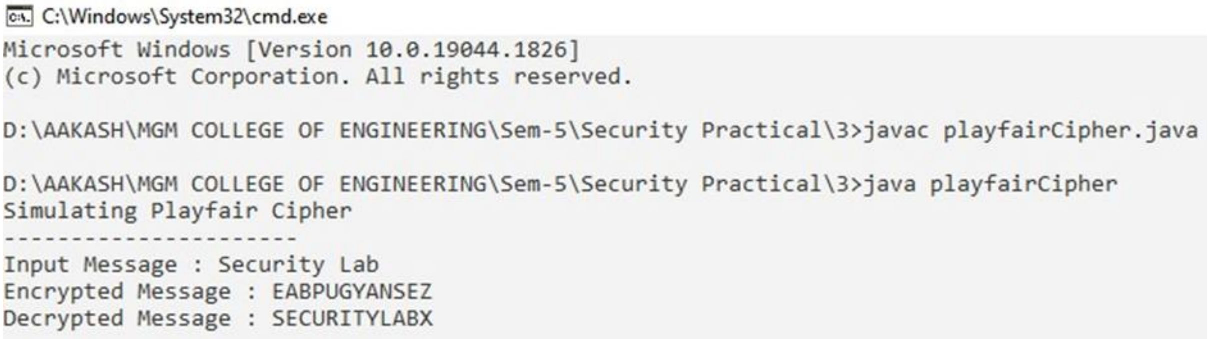


```
C:\Users\Aakas>cd desktop

C:\Users\Aakas\Desktop>javac ProductCipher.java

C:\Users\Aakas\Desktop>java ProductCipher
Enter the input to be encrypted:
The quick brown fox jumps over the lazy dog.
Enter a number:
7

Substituted text:
Ymj%vznhp%gwt|s%kt}%ozrux%t{jw%ymj%qf⌂~%itl3

Transposition Matrix:
Yhszjql
mp%rwf3
j%ku%⌂/
%gtxy~/
vw}%m%/
zt%tji/
n|o{%t/

Final encrypted text:
Yhszjqlmp%rwf3j%ku%⌂/%gtxy~/vw}%m%/zt%tji/n|o{%t/

Plaintext:
The quick brown fox jumps over the lazy dog.*****

C:\Users\Aakas\Desktop>
```

3. Implementation of Cryptanalysis Playfair cipher.

```
import java.awt.Point;
class playfairCipher {
 private static char[][] charTable;
 private static Point[] positions;
 private static String prepareText(String s, boolean chgJtoI) {
 s = s.toUpperCase().replaceAll("[^A-Z]", "");
 return chgJtoI ? s.replace("J", "I") : s.replace("Q", "");
 }
 private static void createTbl(String key, boolean chgJtoI) {
 charTable = new char[5][5];
 positions = new Point[26];
 String s = prepareText(key + "ABCDEFGHIJKLMNOPQRSTUVWXYZ",
chgJtoI);
 int len = s.length();
 for (int i = 0, k = 0; i < len; i++) {
 char c = s.charAt(i);
if (positions[c - 'A'] == null) {
 charTable[k / 5][k % 5] = c;
 positions[c - 'A'] = new Point(k % 5, k / 5);
 k++;
 }
 }
 }
 private static String codec(StringBuilder txt, int dir) {
 int len = txt.length();
 for (int i = 0; i < len; i += 2) {
 char a = txt.charAt(i);
 char b = txt.charAt(i + 1);
 int row1 = positions[a - 'A'].y;
 int row2 = positions[b - 'A'].y;
 int col1 = positions[a - 'A'].x;
 int col2 = positions[b - 'A'].x;
 if (row1 == row2) {
 col1 = (col1 + dir) % 5;
 col2 = (col2 + dir) % 5;
 } else if (col1 == col2) {
 row1 = (row1 + dir) % 5;
 row2 = (row2 + dir) % 5;
 } else {
 int tmp = col1;
 col1 = col2;
 col2 = tmp;
 }
 txt.setCharAt(i, charTable[row1][col1]);
 txt.setCharAt(i + 1, charTable[row2][col2]);
 }
 return txt.toString();
 }
 private static String encode(String s) {
 StringBuilder sb = new StringBuilder(s);
 for (int i = 0; i < sb.length(); i += 2) {
 if (i == sb.length() - 1) {
   sb.append(sb.length() % 2 == 1 ? 'X' : "");
 } else if (sb.charAt(i) == sb.charAt(i + 1)) {
 sb.insert(i + 1, 'X');
 }
 }
 return codec(sb, 1);
 }
 private static String decode(String s) {
 return codec(new StringBuilder(s), 4);
 }
 public static void main(String[] args) throws java.lang.Exception {
 String key = "CSE";
 String txt = "Security Lab"; /* make sure string length is even */ /* change J
to I */
 boolean chgJtoI = true;
 createTbl(key, chgJtoI);
 String enc = encode(prepareText(txt, chgJtoI));
 System.out.println("Simulating Playfair Cipher\n----------------------");
 System.out.println("Input Message : " + txt);
 System.out.println("Encrypted Message : " + enc);
 System.out.println("Decrypted Message : " + decode(enc));
 }
}
```

Output:



```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19044.1826]
(c) Microsoft Corporation. All rights reserved.

D:\AAKASH\MGM COLLEGE OF ENGINEERING\Sem-5\Security Practical\3>javac playfairCipher.java

D:\AAKASH\MGM COLLEGE OF ENGINEERING\Sem-5\Security Practical\3>java playfairCipher
Simulating Playfair Cipher
----------------------
Input Message : Security Lab
Encrypted Message : EABPUGYANSEZ
Decrypted Message : SECURITYLABX
```

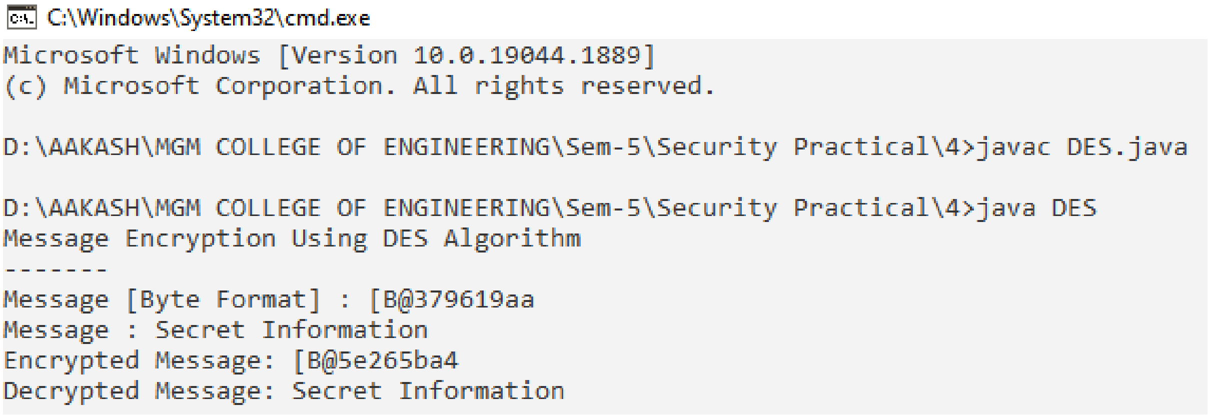
4. Encrypt long messages using various modes of operation using DES.

DES.java

```
import java.security.InvalidKeyException;
import java.security.NoSuchAlgorithmException;
import javax.crypto.BadPaddingException;
import javax.crypto.Cipher;
import javax.crypto.IllegalBlockSizeException;
import javax.crypto.KeyGenerator;
import javax.crypto.NoSuchPaddingException;
import javax.crypto.SecretKey;
public class DES
```

```
{
public static void main(String[] argv) {
try{
 System.out.println("Message Encryption Using DES Algorithm\n-------");  KeyGenerator keygenerator
= KeyGenerator.getInstance("DES");  SecretKey myDesKey = keygenerator.generateKey();
 Cipher desCipher;
 desCipher = Cipher.getInstance("DES/ECB/PKCS5Padding");  desCipher.init(Cipher.ENCRYPT_MODE,
myDesKey);
 byte[] text = "Secret Information ".getBytes();
 System.out.println("Message [Byte Format] : " + text);
 System.out.println("Message : " + new String(text));
 byte[] textEncrypted = desCipher.doFinal(text);
 System.out.println("Encrypted Message: " + textEncrypted);  desCipher.init(Cipher.DECRYPT_MODE,
myDesKey);
 byte[] textDecrypted = desCipher.doFinal(textEncrypted);  System.out.println("Decrypted Message:
" + new
String(textDecrypted));
}catch(NoSuchAlgorithmException e){
e.printStackTrace();
}catch(NoSuchPaddingException e){
e.printStackTrace();
}catch(InvalidKeyException e){
e.printStackTrace();
}catch(IllegalBlockSizeException e){
e.printStackTrace();
}catch(BadPaddingException e){
e.printStackTrace();
}
}
}
```

Output:



```
Microsoft Windows [Version 10.0.19044.1889]
(c) Microsoft Corporation. All rights reserved.

D:\AAKASH\MGM COLLEGE OF ENGINEERING\Sem-5\Security Practical\4>javac DES.java

D:\AAKASH\MGM COLLEGE OF ENGINEERING\Sem-5\Security Practical\4>java DES
Message Encryption Using DES Algorithm
-------
Message [Byte Format] : [B@379619aa
Message : Secret Information
Encrypted Message: [B@5e265ba4
Decrypted Message: Secret Information
```

5. Implementation and analysis of RSA cryptosystem

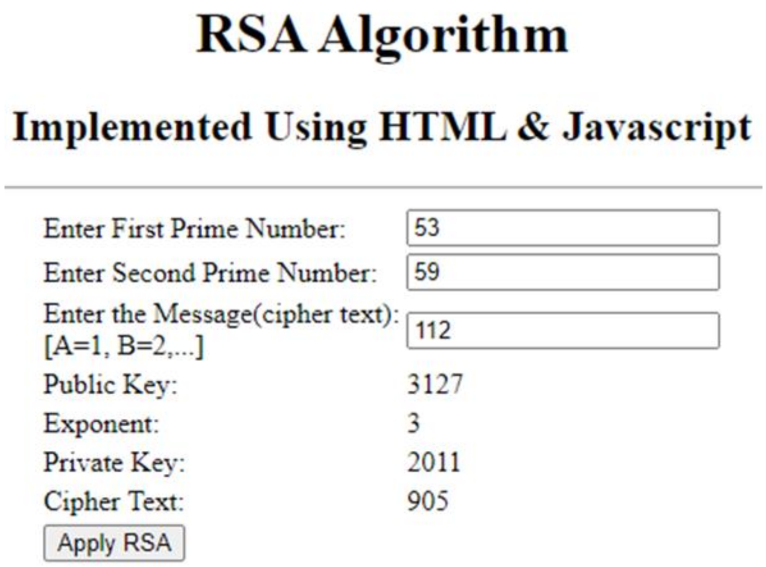
rsa.html

```
<html>
<head>
```

```
<title>RSA Encryption</title>
<meta name="viewport" content="width=device-width, initial-scale=1.0">
</head>
<body>
<center>
<h1>RSA Algorithm</h1>
<h2>Implemented Using HTML & Javascript</h2>
<hr>
<table>
<tr>
<td>Enter First Prime Number:</td>
<td><input type="number" value="53" id="p"></td>
</tr>
<tr>
<td>Enter Second Prime Number:</td>
<td><input type="number" value="59" id="q"></p>

</td>
</tr>
<tr>
<td>Enter the Message(cipher text):<br>[A=1, B=2,...]</td>
<td><input type="number" value="89" id="msg"></p>
</td>
</tr>
<tr>
<td>Public Key:</td>
<td>
<p id="publickey"></p>
</td>
</tr>
<tr>
<td>Exponent:</td>
<td>
<p id="exponent"></p>
</td>
</tr>
<tr>
<td>Private Key:</td>
<td>
<p id="privatekey"></p>
</td>
</tr>
<tr>
<td>Cipher Text:</td>
<td>
<p id="ciphertext"></p>
</td>
</tr>
```

```
<tr>
<td><button onclick="RSA();">Apply RSA</button></td>

</tr>
</table>
</center>
</body>
<script type="text/javascript"> function
RSA() {
var gcd, p, q, no, n, t, e, i, x;

gcd = function (a, b) { return (!b) ? a : gcd(b, a % b); }; p =
document.getElementById('p').value;
q = document.getElementById('q').value;
no = document.getElementById('msg').value; n = p *
q;
t = (p - 1) * (q - 1);
for (e = 2; e < t; e++) { if
(gcd(e, t) == 1) {
break;
}
}
for (i = 0; i < 10; i++) { x =
1 + i * t
if (x % e == 0) { d
= x / e; break;
}
}
ctt = Math.pow(no, e).toFixed(0); ct =
ctt % n;
dtt = Math.pow(ct, d).toFixed(0); dt =
dtt % n;
document.getElementById('publickey').innerHTML = n;
document.getElementById('exponent').innerHTML = e;
document.getElementById('privatekey').innerHTML = d;
document.getElementById('ciphertext').innerHTML = ct;
}
</script>
</html>
```

Output:



RSA Algorithm

Implemented Using HTML & Javascript

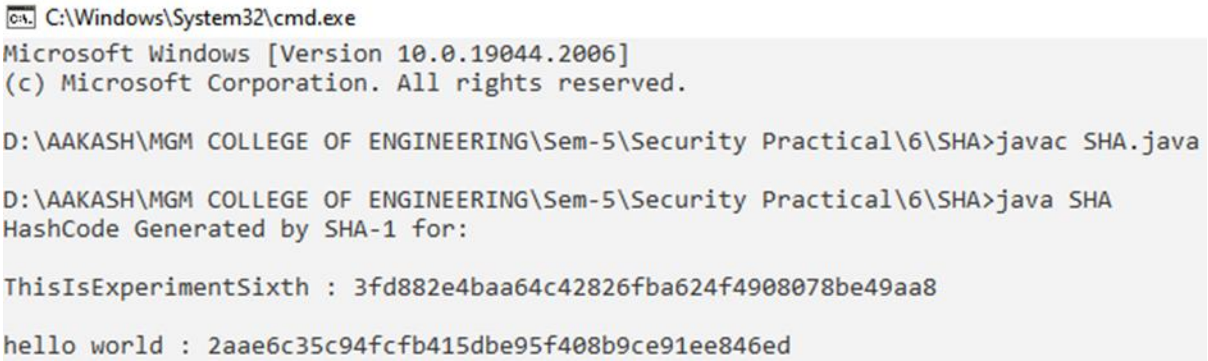
Enter First Prime Number:	53
Enter Second Prime Number:	59
Enter the Message(cipher text): [A=1, B=2,...]	112
Public Key:	3127
Exponent:	3
Private Key:	2011
Cipher Text:	905

6. Implementation of Cryptographic Hash Functions and Applications (HMAC): SHA

SHA.java

```
// Java program to calculate SHA-1 hash value

import java.math.BigInteger;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;

public class SHA {
        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 = "ThisIsExperimentSixth";
                System.out.println("\n" + s1 + " : " + encryptThisString(s1));

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

Output:



```
Microsoft Windows [Version 10.0.19044.2006]
(c) Microsoft Corporation. All rights reserved.

D:\AAKASH\MGM COLLEGE OF ENGINEERING\Sem-5\Security Practical\6\SHA>javac SHA.java

D:\AAKASH\MGM COLLEGE OF ENGINEERING\Sem-5\Security Practical\6\SHA>java SHA
HashCode Generated by SHA-1 for:

ThisIsExperimentSixth : 3fd882e4baa64c42826fba624f4908078be49aa8

hello world : 2aae6c35c94fcfb415dbe95f408b9ce91ee846ed
```