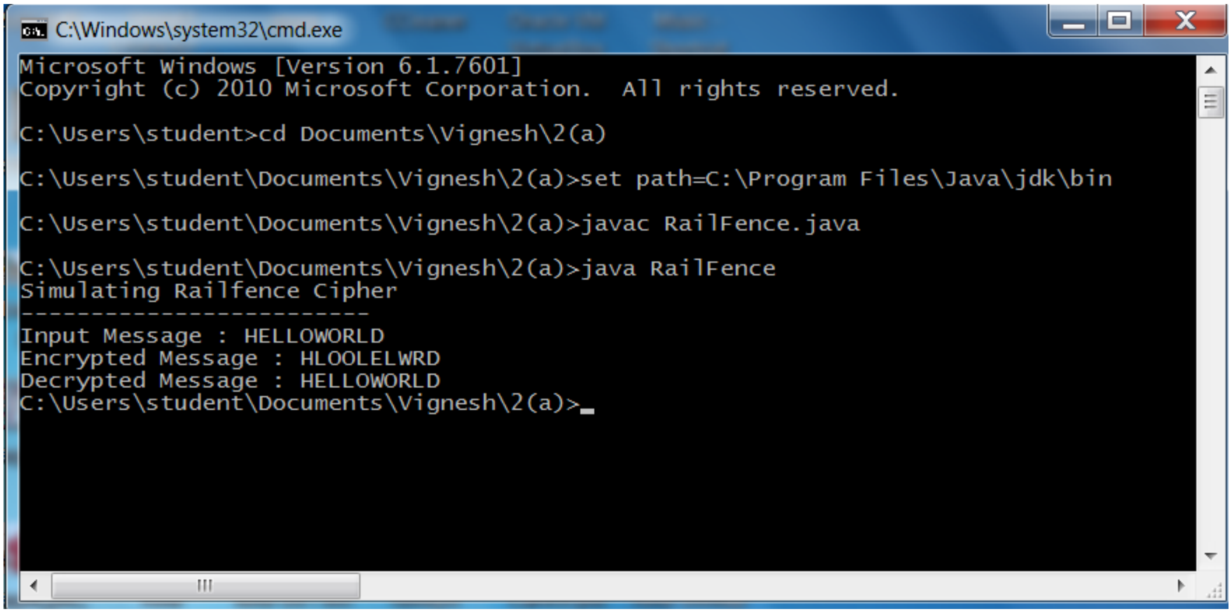
PROGRAM:

railFenceCipher.java

```
class railfenceCipherHelper
{
    int depth;
    String encode(String msg, int depth) throws Exception
    {
        int r = depth;
        int l = msg.length();
        int c = l / depth;
        int k = 0;
        char mat[][] = new char[r][c];
        String enc = "";
        for (int i = 0; i < c; i++)
        {
            for (int j = 0; j < r; j++)
            {
                if (k != l)
                {
                    mat[j][i] = msg.charAt(k++);
                }
                 else
                {
                    mat[j][i] = 'X';
                }
            }
        }
        for (int i = 0; i < r; i++)
        {
            for (int j = 0; j < c; j++)
            {
                enc += mat[i][j];
            }
        }
        return enc;
    }
    String decode(String encmsg, int depth) throws Exception
    {
        int r = depth;
        int l = encmsg.length();
        int c = l / depth; int k = 0;
        char mat[][] = new char[r][c];
        String dec = "";
        for (int i = 0; i < r; i++)
        {
            for (int j = 0; j < c; j++)
            {
                mat[i][j] = encmsg.charAt(k++);
            }
        }
        for (int i = 0; i < c; i++)
        {
            for (int j = 0; j < r; j++)
            {
                dec += mat[j][i];
            }
        }
        return dec;
    }
}
class railFenceCipher
{
    public static void main(String[] args) throws java.lang.Exception
    {
        railfenceCipherHelper rf = new railfenceCipherHelper();
        String msg, enc, dec;
        msg = "HelloWorld";
        int depth = 2;
        enc = rf.encode(msg,depth);
        dec = rf.decode(enc, depth);
        System.out.println("Simulating Railfence Cipher\n ");
        System.out.println("Input Message: " + msg);
        System.out.println("Encrypted Message: " + enc);
        System.out.printf("Decrypted Message: " + dec);
    }
}
```

Output:

Simulating Railfence Cipher
Input Message: HELLOWORLD
Encrypted Message: HLOOLELWRD
Decrypted Message: HELLOWORLD



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2010 Microsoft Corporation.  All rights reserved.

C:\Users\student>cd Documents\Vignesh\2(a)

C:\Users\student\Documents\Vignesh\2(a)>set path=C:\Program Files\Java\jdk\bin

C:\Users\student\Documents\Vignesh\2(a)>javac RailFence.java

C:\Users\student\Documents\Vignesh\2(a)>java RailFence
Simulating Railfence Cipher
-------------------------
Input Message : HELLOWORLD
Encrypted Message : HLOOLELWRD
Decrypted Message : HELLOWORLD
C:\Users\student\Documents\Vignesh\2(a)>_
```

PROGRAM:

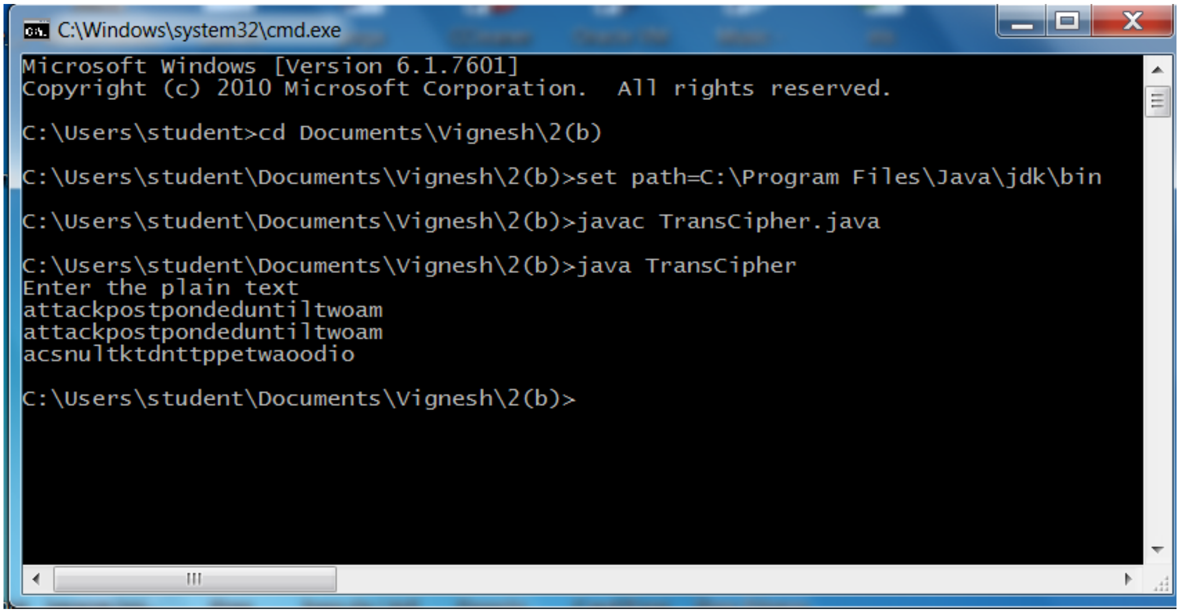
TransCipher.java

```
import java.util.*;
class TransCipher
{
        public static void main(String args[])
        {
                Scanner sc = new Scanner(System.in);
                System.out.println("Enter the plain text");
                String pl = sc.nextLine();
                sc.close();
                String s = "";
                int start = 0;
                for (int i = 0; i < pl.length(); i++)
                {
                        if (pl.charAt(i) == ' ')
                        {
                                s = s + pl.substring(start, i);
                                start = i + 1;
                        }
                }
                s = s + pl.substring(start);
                System.out.print(s);
                System.out.println();
                 // end of space deletion
                int k = s.length();
                int l = 0;
                int col = 4;
                int row = s.length() / col;
                char ch[][] = new char[row][col];
                for (int i = 0; i < row; i++)
                {
                for (int j = 0; j < col; j++)
                {
```

```
            if (l < k)
            {
                ch[i][j] = s.charAt(l);
                l++;
            }
            else
            {
                ch[i][j] = '#';
            }
        }
    }
    // arranged in matrix
    char trans[][] = new char[col][row];
    for (int i = 0; i < row; i++)
    {
        for (int j = 0; j < col; j++)
        {
            trans[j][i] = ch[i][j];
        }
    }
    for (int i = 0; i < col; i++)
    {
        for (int j = 0; j < row; j++)
        {
            System.out.print(trans[i][j]);
        }
    }
    // display
    System.out.println();
     }
}
```

Output:

Enter the plain text
attactpostponeduntiltwoam
attackpostpondeduntiltwoam
acsnultktdnttppetwaoodio



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2010 Microsoft Corporation.  All rights reserved.

C:\Users\student>cd Documents\Vignesh\2(b)

C:\Users\student\Documents\Vignesh\2(b)>set path=C:\Program Files\Java\jdk\bin

C:\Users\student\Documents\Vignesh\2(b)>javac TransCipher.java

C:\Users\student\Documents\Vignesh\2(b)>java TransCipher
Enter the plain text
attackpostpondeduntiltwoam
attackpostpondeduntiltwoam
acsnultktdnttppetwaoodio

C:\Users\student\Documents\Vignesh\2(b)>
```

PROGRAM:

DES.java

```
import javax.swing.*;
import java.security.SecureRandom;
import javax.crypto.Cipher;
import javax.crypto.KeyGenerator;
import javax.crypto.SecretKey;
import javax.crypto.spec.SecretKeySpec;
import java.util.Random ;
class DES
{
    byte[] skey = new byte[1000];
    String skeyString;
    static byte[] raw;
    String inputMessage,encryptedData,decryptedMessage;
    public DES()
    {
        try
        {
            generateSymmetricKey();
            inputMessage=JOptionPane.showInputDialog(null,"Enter message to encrypt");
            byte[] ibyte = inputMessage.getBytes();
            byte[] ebyte=encrypt(raw, ibyte);
            String encryptedData = new String(ebyte);
            System.out.println("Encrypted message "+encryptedData);
            JOptionPane.showMessageDialog(null,"Encrypted Data"+"\n"+ encryptedData);
            byte[] dbyte= decrypt(raw,ebyte);
            String decryptedMessage = new String(dbyte);
            System.out.println("Decrypted message "+decryptedMessage);
            JOptionPane.showMessageDialog(null,"Decrypted Data"+"\n"+ decryptedMessage);
        }
        catch(Exception e)
        {
            System.out.println(e);
        }
    }
    void generateSymmetricKey()
    {
        try
        {
            Random r = new Random();
            int num = r.nextInt(10000);
            String knum = String.valueOf(num);
            byte[] knumb = knum.getBytes();
            skey=getRawKey(knumb);
            skeyString = new String(skey);
            System.out.println("DES Symmetric key = "+skeyString);
        }
        catch(Exception e)
        {
            System.out.println(e);
        }
    }
    private static byte[] getRawKey(byte[] seed) throws Exception
    {
        KeyGenerator kgen = KeyGenerator.getInstance("DES");
        SecureRandom sr = SecureRandom.getInstance("SHA1PRNG");
        sr.setSeed(seed);
        kgen.init(56, sr);
        SecretKey skey = kgen.generateKey();
        raw = skey.getEncoded();
        return raw;
    }
    private static byte[] encrypt(byte[] raw, byte[] clear) throws Exception
    {
        SecretKeySpec skeySpec = new SecretKeySpec(raw, "DES");
```

```
                Cipher cipher = Cipher.getInstance("DES");
                cipher.init(Cipher.ENCRYPT_MODE, skeySpec);
                byte[] encrypted = cipher.doFinal(clear);
                return encrypted;
        }
        private static byte[] decrypt(byte[] raw, byte[] encrypted) throws Exception
        {
                SecretKeySpec skeySpec = new SecretKeySpec(raw, "DES");
                Cipher cipher = Cipher.getInstance("DES");
                cipher.init(Cipher.DECRYPT_MODE, skeySpec);
                byte[] decrypted = cipher.doFinal(encrypted);
                return decrypted;
        }
        public static void main(String args[])
        {
                DES des = new DES();
        }
}
```

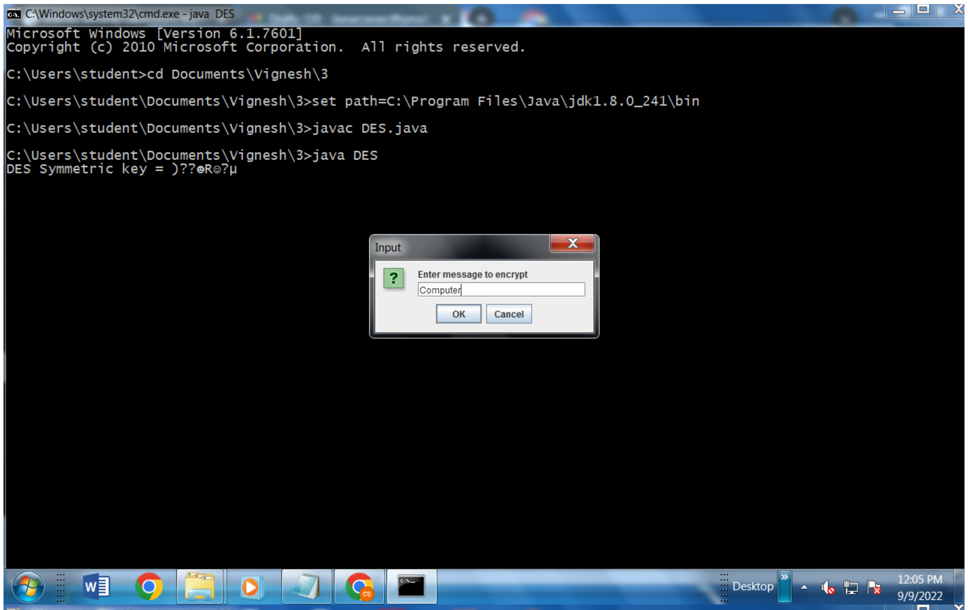
OUTPUT:

Message Encryption Using DES Algorithm

DES Symmetric key : uz/^_!0c>

Encrypted message : j#^$€€?\e#->

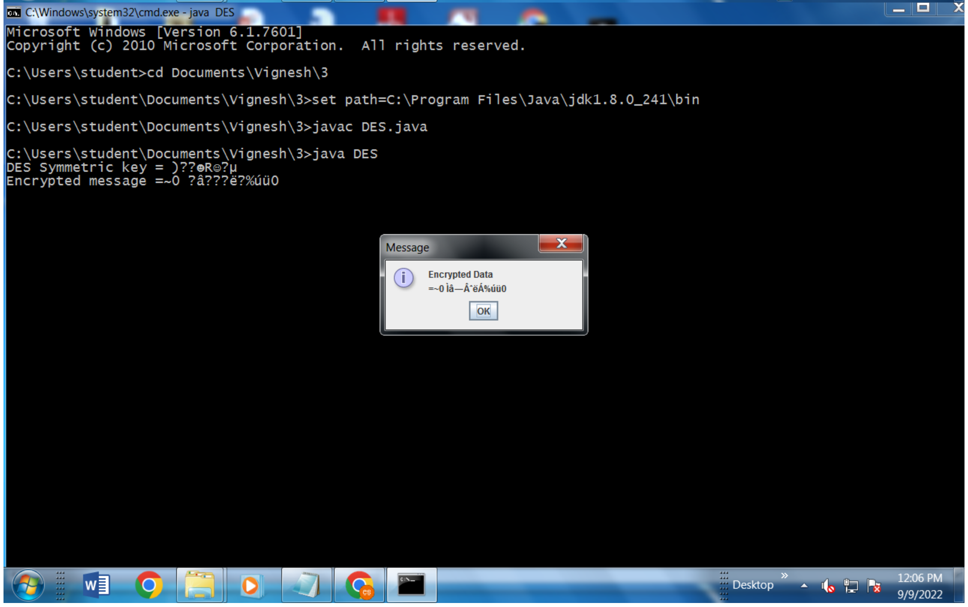
Decrypted message : computer



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2010 Microsoft Corporation.  All rights reserved.

C:\Users\student>cd Documents\Vignesh\3

C:\Users\student\Documents\Vignesh\3>set path=C:\Program Files\Java\jdk1.8.0_241\bin

C:\Users\student\Documents\Vignesh\3>javac DES.java

C:\Users\student\Documents\Vignesh\3>java DES
DES Symmetric key = )??☻R☺?µ
```

Enter message to encrypt

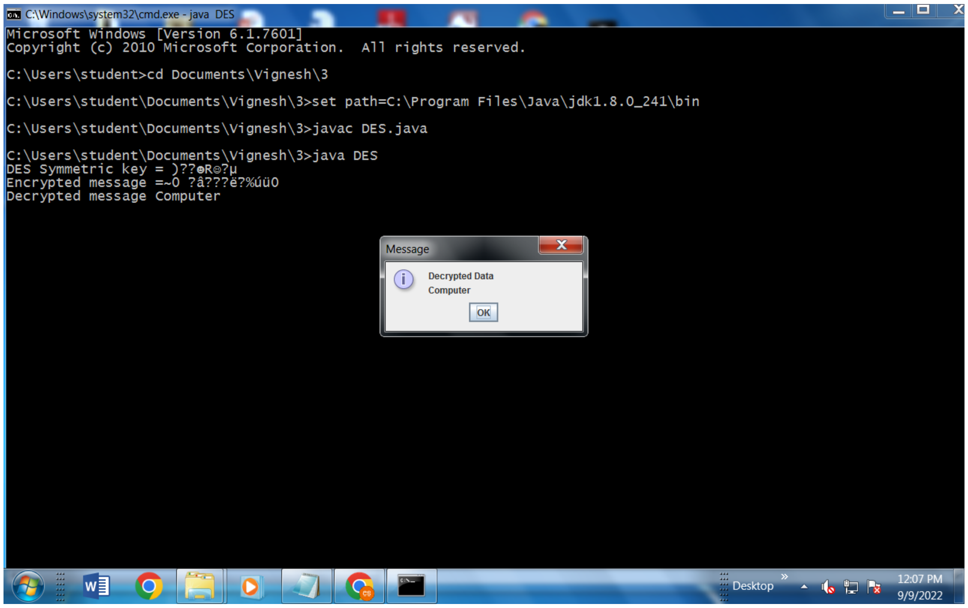
Computer



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2010 Microsoft Corporation.  All rights reserved.

C:\Users\student>cd Documents\Vignesh\3

C:\Users\student\Documents\Vignesh\3>set path=C:\Program Files\Java\jdk1.8.0_241\bin

C:\Users\student\Documents\Vignesh\3>javac DES.java

C:\Users\student\Documents\Vignesh\3>java DES
DES Symmetric key = )??☻R☺?µ
Encrypted message =~0 ?â???ë?%úü0
```

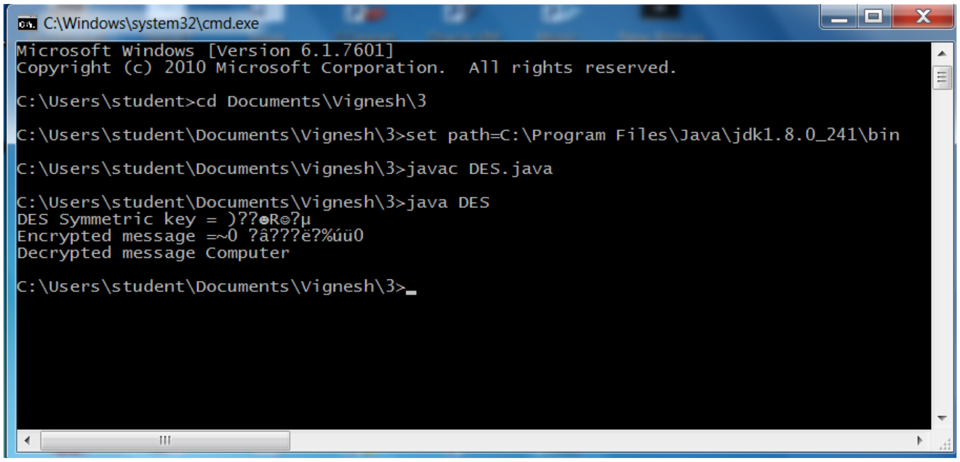
Encrypted Data

=~0 Ìâ—Âˆ ëÁ%úü0



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2010 Microsoft Corporation.  All rights reserved.

C:\Users\student>cd Documents\Vignesh\3

C:\Users\student\Documents\Vignesh\3>set path=C:\Program Files\Java\jdk1.8.0_241\bin

C:\Users\student\Documents\Vignesh\3>javac DES.java

C:\Users\student\Documents\Vignesh\3>java DES
DES Symmetric key = )??☻R☺?µ
Encrypted message =~0 ?â???ë?%úü0
Decrypted message Computer
```

Decrypted Data
Computer



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2010 Microsoft Corporation.  All rights reserved.

C:\Users\student>cd Documents\Vignesh\3

C:\Users\student\Documents\Vignesh\3>set path=C:\Program Files\Java\jdk1.8.0_241\bin

C:\Users\student\Documents\Vignesh\3>javac DES.java

C:\Users\student\Documents\Vignesh\3>java DES
DES Symmetric key = )??☻R☺?µ
Encrypted message =~0 ?â???ë?%úü0
Decrypted message Computer

C:\Users\student\Documents\Vignesh\3>_
```

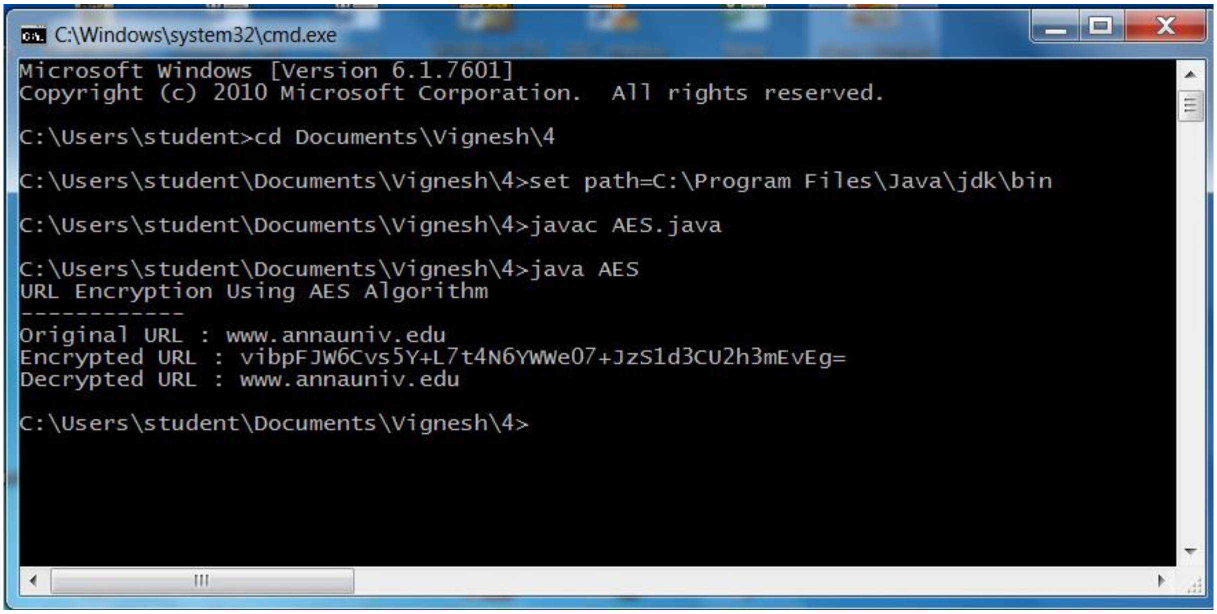
PROGRAM:

AES.java

```
import java.io.UnsupportedEncodingException;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
import java.util.Arrays;
import java.util.Base64;
import javax.crypto.Cipher;
import javax.crypto.spec.SecretKeySpec;


public class AES
{
    private static SecretKeySpec secretKey;
    private static byte[] key;
    public static void setKey(String myKey)
    {
        MessageDigest sha = null;
        try
        {
            key = myKey.getBytes("UTF-8");
            sha = MessageDigest.getInstance("SHA-1");
            key = sha.digest(key);
            key = Arrays.copyOf(key, 16);
            secretKey = new SecretKeySpec(key, "AES");
        }
        catch (NoSuchAlgorithmException e)
        {
            e.printStackTrace();
        }
        catch (UnsupportedEncodingException e)
        {
            e.printStackTrace();
        }
    }
    public static String encrypt(String strToEncrypt, String secret)
    {
        try
        {
            setKey(secret);
            Cipher cipher = Cipher.getInstance("AES/ECB/PKCS5Padding");
            cipher.init(Cipher.ENCRYPT_MODE, secretKey);
            return Base64.getEncoder().encodeToString(cipher.doFinal(strToEncr
ypt.getBytes("UTF-8")));
        }
        catch (Exception e)
        {
            System.out.println("Error while encrypting: " + e.toString());
        }
        return null;
    }

    public static String decrypt(String strToDecrypt, String secret)
    {
        try
        {
            setKey(secret);
            Cipher cipher = Cipher.getInstance ("AES/ECB/PKCS5PADDING");
            cipher.init(Cipher.DECRYPT_MODE, secretKey);
            return new String(cipher.doFinal(Base64.get
Decoder().decode(strToDecrypt)));
        }

        catch (Exception e)
        {
            System.out.println("Error while decrypting: " + e.toString());
        }
        return null;
    }


    public static void main(String[] args)
    {
        final String secretKey = "annaUniversity";
        String originalString = "www.annauniv.edu";
        String encryptedString = AES.encrypt(originalString, secretKey);
        String decryptedString = AES.decrypt(encryptedString, secretKey);
        System.out.println("URL Encryption Using AES Algorithm\n------ ");
        System.out.println("Original URL : " + originalString);
        System.out.println("Encrypted URL : " + encryptedString);
        System.out.println("Decrypted URL : " + decryptedString);
    }
}
```

OUTPUT:

URL Encryption Using AES Algorithm
Original URL : www.annauniv.edu
Encrypted URL : vibpFJW6Cvs5Y+L7t4N6YWWe07+JzS1d3CU2h3mEvEg=
Decrypted URL : www.annauniv.edu



```
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2010 Microsoft Corporation.  All rights reserved.

C:\Users\student>cd Documents\Vignesh\4

C:\Users\student\Documents\Vignesh\4>set path=C:\Program Files\Java\jdk\bin

C:\Users\student\Documents\Vignesh\4>javac AES.java

C:\Users\student\Documents\Vignesh\4>java AES
URL Encryption Using AES Algorithm
------------
Original URL : www.annauniv.edu
Encrypted URL : vibpFJW6Cvs5Y+L7t4N6YWWe07+JzS1d3CU2h3mEvEg=
Decrypted URL : www.annauniv.edu

C:\Users\student\Documents\Vignesh\4>
```

Program:

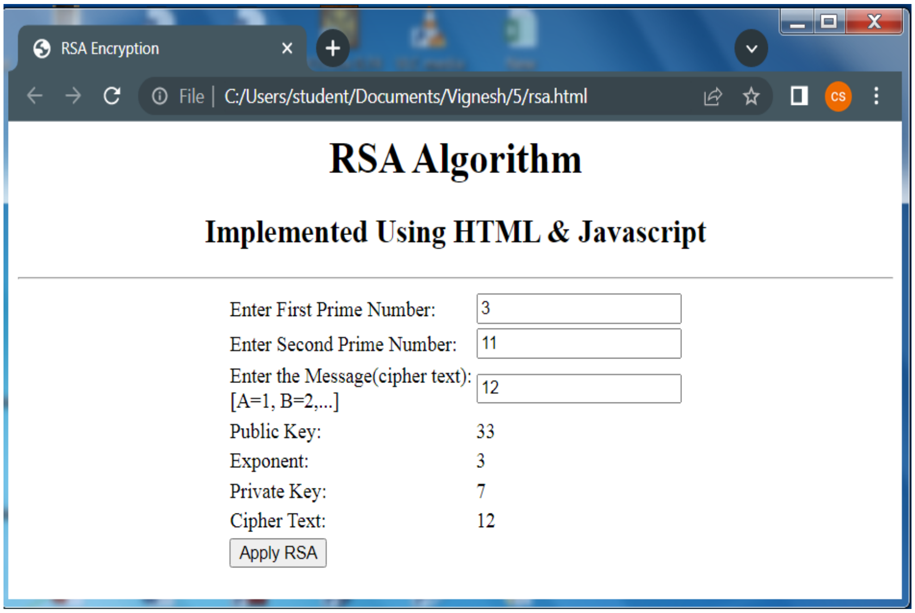
rsa.html

```
<html>
<head>
    <title>RSA Encryption</title>
    <meta name="viewport" content="width=device-width, initiascale=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);
        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: 3

Enter Second Prime Number: 11

Enter the Message(cipher text): [A=1, B=2,...] 12

Public Key: 33

Exponent: 3

Private Key: 7

Cipher Text: 12