Ex.No: 1(i) IMPLEMENTATION OF CAESAR CIPHER

AIM

To write a Java Program for implementing Caesar Cipher.

ALGORITHM

Step 1: Start the program.

Step 2: Define a class CaesarCipher.

Step 3: Declare a string ALPHABET.

Step 4: Define a function encrypt() to produce a cipher text and decrypt() to reproduce the plain text.

Step 5: Define a main(), get the string and call encrypt() to encrypt the string and decrypt() to reproduce the plain text and display it.

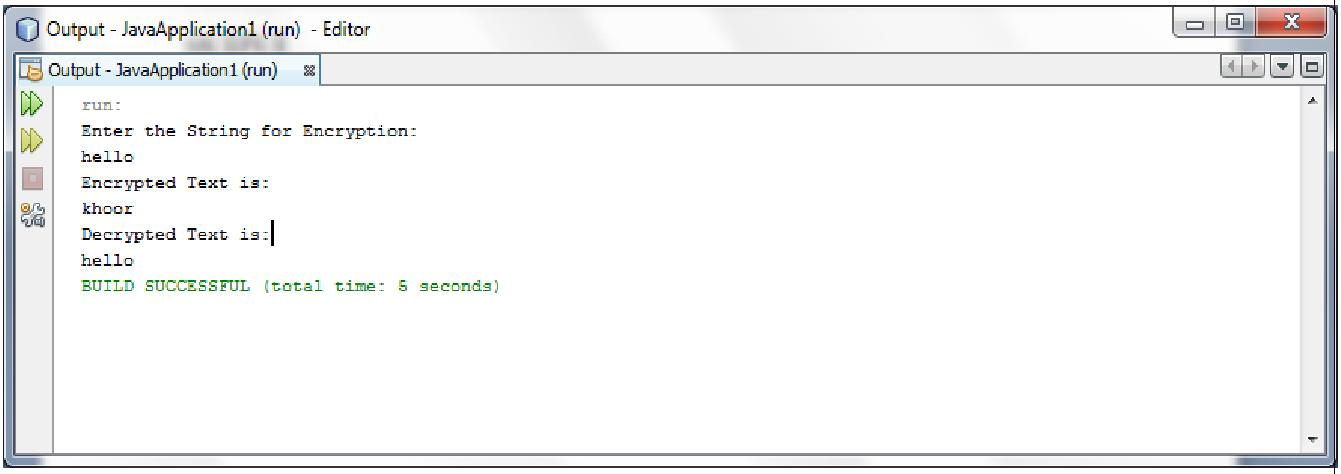
Step 6: Stop the program.

PROGRAM

```
package javaapplication1;
import java.util.Scanner;
public class CaesarCipher
{
	public static final String ALPHABET = "abcdefghijklmnopqrstuvwxyz";
	public static String encrypt(String plainText, int shiftKey)
	{
		plainText = plainText.toLowerCase();
		String cipherText = "";
		for (int i = 0; i < plainText.length(); i++)
		{
			int charPosition = ALPHABET.indexOf(plainText.charAt(i));
			int keyVal = (shiftKey + charPosition) % 26;
			char replaceVal = ALPHABET.charAt(keyVal);
			cipherText += replaceVal;
		}
		return cipherText;
	}
```

```
    public static String decrypt(String cipherText, int shiftKey)
    {
        cipherText = cipherText.toLowerCase();
        String plainText = "";
        for (int i = 0; i < cipherText.length(); i++)
        {
            int charPosition = ALPHABET.indexOf(cipherText.charAt(i));
            int keyVal = (charPosition - shiftKey) % 26;
            if (keyVal < 0)
            {
                keyVal = ALPHABET.length() + keyVal;
            }
            char replaceVal = ALPHABET.charAt(keyVal);
            plainText += replaceVal;
        }
        return plainText;
    }
    public static void main(String[] args)
    {
        try (Scanner sc = new Scanner(System.in))
        {
            System.out.println("Enter the String for Encryption: ");
            String message = new String();
            message = sc.next();
            System.out.println("Encrypted Text is:");
            System.out.println(encrypt(message, 3));
            System.out.println("Decrypted Text is:");
            System.out.println(decrypt(encrypt(message, 3), 3));
        }
    }
}
```

OUTPUT



Output - JavaApplication1 (run) - Editor
Output - JavaApplication1 (run)

```
run:
Enter the String for Encryption:
hello
Encrypted Text is:
khoor
Decrypted Text is:
hello
BUILD SUCCESSFUL (total time: 5 seconds)
```

RESULT

Thus java program to implement Caesar Cipher was written, executed and output is verified successfully.

Ex.No: 1(ii) IMPLEMENTATION OF PLAYFAIR CIPHER

AIM

To write a Java program for implementing Playfair Cipher.

ALGORITHM

Step 1: Start the program.

Step 2: Define a class Basic to find the index of a char.

Step 3: Define a class PlayFair to define the key matrix, find the row position, column position, encrypt the text and decrypt the text.

Step 4: Define a class PlayFairCipher, to get the plain text and then to encrypt and decrypt the text.

Step 5: Display the encrypted text and decrypted text.

Step 6: Stop the program.

PROGRAM

```
package javaapplication1;
import java.util.*;
class Basic
{
        String allChar="ABCDEFGHIJKLMNOPQRSTUVWXYZ";
        boolean indexOfChar(char c)
        {
                for(int i=0;i < allChar.length();i++)
                {
                        if(allChar.charAt(i)==c)
                                return true;
                }
                return false;
        }
}
 class PlayFair
{
        Basic b=new Basic();
        char keyMatrix[][]=new char[5][5];
```

```
boolean repeat(char c)
{
    if(!b.indexOfChar(c))
    {
        return true;
    }
    for(int i=0;i < keyMatrix.length;i++)
    {
        for(int j=0;j < keyMatrix[i].length;j++)
        {
            if(keyMatrix[i][j]==c || c=='J')
                return true;
        }
    }
    return false;
}
void insertKey(String key)
{
    key=key.toUpperCase();
    key=key.replaceAll("J", "I");
    key=key.replaceAll(" ", "");
    int a=0,b=0;
    for(int k=0;k < key.length();k++)
    {
        if(!repeat(key.charAt(k)))
        {
            keyMatrix[a][b++]=key.charAt(k);
            if(b>4)
            {
                b=0;
                a++;
            }
        }
    }
    char p='A';
    while(a < 5)
    {
        while(b < 5)
        {
            if(!repeat(p))
            {
                keyMatrix[a][b++]=p;
            }
            p++;
        }
        b=0;
        a++;
    }
    System.out.print("-------------------------Key Matrix-------------------");
    for(int i=0;i < 5;i++)
    {
        System.out.println();
        for(int j=0;j < 5;j++)
            System.out.print("\t"+keyMatrix[i][j]);
    }
    System.out.println("\n---------------------------------------------------------");
}
int rowPos(char c)
{
    for(int i=0;i < keyMatrix.length;i++)
    {
        for(int j=0;j < keyMatrix[i].length;j++)
        {
            if(keyMatrix[i][j]==c)
                return i;
        }
    }
     return -1;
}
```

```
int columnPos(char c)
{
    for(int i=0;i < keyMatrix.length;i++)
    {
        for(int j=0;j < keyMatrix[i].length;j++)
        {
            if(keyMatrix[i][j]==c)
                return j;
        }
    }
    return -1;
}
String encryptChar(String plain)
{
    plain=plain.toUpperCase();
    char a=plain.charAt(0),b=plain.charAt(1);
    String cipherChar="";
    int r1,c1,r2,c2;
    r1=rowPos(a);
    c1=columnPos(a);
    r2=rowPos(b);
    c2=columnPos(b);
    if(c1==c2)
    {
        ++r1;
        ++r2;
        if(r1>4)
            r1=0;
        if(r2>4)
            r2=0;
        cipherChar+=keyMatrix[r1][c2];
        cipherChar+=keyMatrix[r2][c1];
    }
    else if(r1==r2)
    {
```

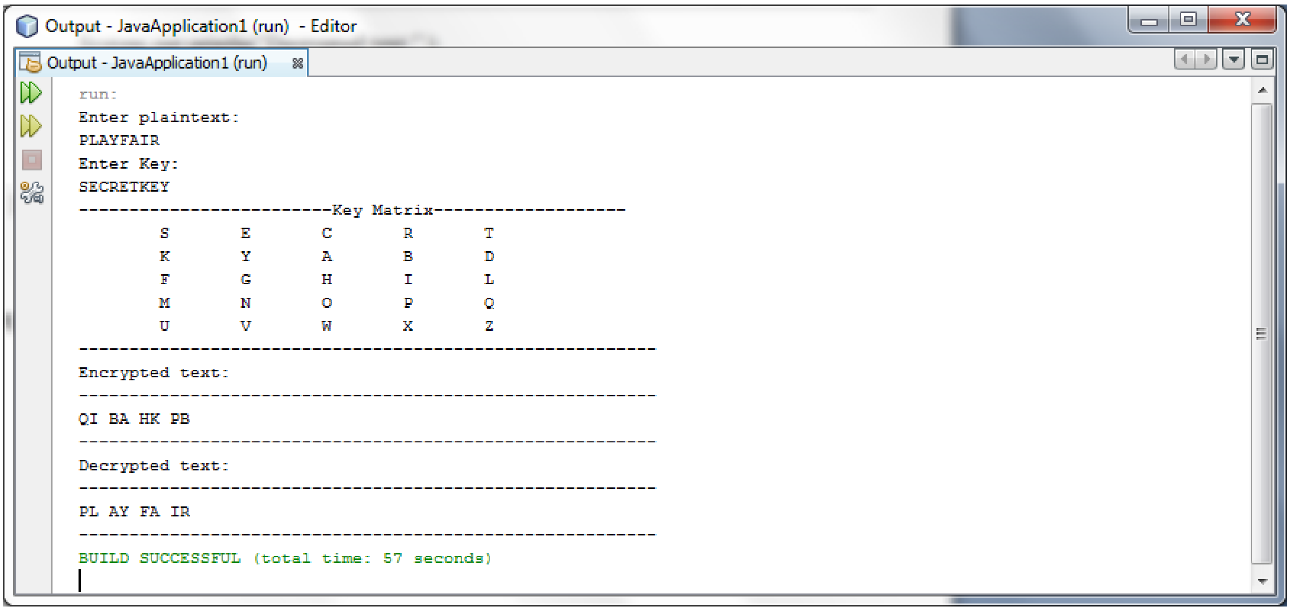
```
                ++c1;
                ++c2;
                if(c1>4)
                        c1=0;
                if(c2>4)
                        c2=0;
                cipherChar+=keyMatrix[r1][c1];
                cipherChar+=keyMatrix[r2][c2];
        }
        else
        {
                cipherChar+=keyMatrix[r1][c2];
                cipherChar+=keyMatrix[r2][c1];
        }
        return cipherChar;
}
String Encrypt(String plainText,String key)
{
        insertKey(key);
        String cipherText="";
        plainText=plainText.replaceAll("j", "i");
        plainText=plainText.replaceAll(" ", "");
        plainText=plainText.toUpperCase();
        int len=plainText.length();
        if(len/2!=0)
        {
                plainText+="X";
                ++len;
        }
        for(int i=0;i < len-1;i=i+2)
        {
                cipherText+=encryptChar(plainText.substring(i,i+2));
                cipherText+=" ";
        }
        return cipherText;      }
```

```
String decryptChar(String cipher)
{
    cipher=cipher.toUpperCase();
    char a=cipher.charAt(0),b=cipher.charAt(1);
    String plainChar="";
    int r1,c1,r2,c2;
    r1=rowPos(a);
    c1=columnPos(a);
    r2=rowPos(b);
    c2=columnPos(b);
    if(c1==c2)
    {
        --r1;
        --r2;
        if(r1 < 0)
            r1=4;
        if(r2 < 0)
            r2=4;
        plainChar+=keyMatrix[r1][c2];
        plainChar+=keyMatrix[r2][c1];
    }
    else if(r1==r2)
    {
        --c1;
        --c2;
        if(c1 < 0)
            c1=4;
        if(c2 < 0)
            c2=4;
        plainChar+=keyMatrix[r1][c1];
        plainChar+=keyMatrix[r2][c2];
    }
    else
    {
        plainChar+=keyMatrix[r1][c2];
```

```
                plainChar+=keyMatrix[r2][c1];
            }
            return plainChar;
        }
        String Decrypt(String cipherText,String key)
        {
            String plainText="";
            cipherText=cipherText.replaceAll("j", "i");
            cipherText=cipherText.replaceAll(" ", "");
            cipherText=cipherText.toUpperCase();
            int len=cipherText.length();
            for(int i=0;i < len-1;i=i+2)
            {
                plainText+=decryptChar(cipherText.substring(i,i+2));
                plainText+=" ";
            }
            return plainText;
        }
    }
    class PlayfairCipher
    {
        public static void main(String args[])throws Exception
        {
            PlayFair p=new PlayFair();
            Scanner scn=new Scanner(System.in);
            String key,cipherText,plainText;
            System.out.println("Enter plaintext:");
            plainText=scn.nextLine();
            System.out.println("Enter Key:");
            key=scn.nextLine();
            cipherText=p.Encrypt(plainText,key);
            System.out.println("Encrypted text:");
            System.out.println("---------------------------------------------------------\n"+cipherText);
             System.out.println("---------------------------------------------------------");
            String encryptedText=p.Decrypt(cipherText, key);
```

```
                System.out.println("Decrypted text:" );
                System.out.println("-------------------------------------------------------
                \n"+encryptedText);
                System.out.println("-------------------------------------------------------");
        }
}
```

OUTPUT



Output - JavaApplication1 (run) - Editor
Output - JavaApplication1 (run)

```
run:
Enter plaintext:
PLAYFAIR
Enter Key:
SECRETKEY
-------------------------Key Matrix-------------------
        S       E       C       R       T
        K       Y       A       B       D
        F       G       H       I       L
        M       N       O       P       Q
        U       V       W       X       Z
---------------------------------------------------------
Encrypted text:
---------------------------------------------------------
QI BA HK PB
---------------------------------------------------------
Decrypted text:
---------------------------------------------------------
PL AY FA IR
---------------------------------------------------------
BUILD SUCCESSFUL (total time: 57 seconds)
```

RESULT

Thus java program to implement PlayFair Cipher was written, executed and output is verified successfully.

Ex.No: 1(iii) IMPLEMENTATION OF HILL CIPHER

AIM

To write a Java Program for implementing Hill Cipher.

ALGORITHM

Step 1: Start the program.

Step 2: Define a class HillCipher, in that declare 2 array one for key, other for inverse key and define a string key.

Step 3: In this class, define a main(), get the choice to encrypt or decrypt.

Step 4: Based on the choice call encrypt() and decrypt() to find cipher and plain text.

Step 5: Display the result.

Step 6: Stop the program.

PROGRAM

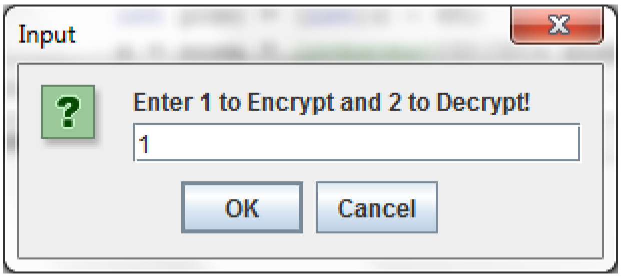
```
package javaapplication1;
import javax.swing.JOptionPane;
public class HillCipher
{
        //the 3x3 key matrix for 3 characters at once
        public static int[][] keymat = new int[][]{
           { 1, 2, 1 },
           { 2, 3, 2 },
           { 2, 2, 1 },
        };
        public static int[][] invkeymat = new int[][]{
           { -1, 0, 1 },
           { 2, -1, 0 },
           { -2, 2, -1},
        };
        public static String key = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
        public static void main(String[] args)
        { // TODO code application logic here
                String text,outtext ="";
                int ch, n;
```

```
    ch = Integer.parseInt(JOptionPane.showInputDialog(null, "Enter 1 to Encrypt and 2 to Decrypt!"));
    text = JOptionPane.showInputDialog(null, "Enter plain/cipher text to encrypt?");
    text = text.toUpperCase();
    text = text.replaceAll("\\s",""); //removing spaces
    n = text.length() % 3;
    if(n!=0)
    {
        for(int i = 1; i<= (3-n);i++)
        {
            text+= 'X';
        }
    }
    System.out.println("Padded Text:" + text);
    char[] ptextchars = text.toCharArray();
    switch(ch)
    {
        case 1:
        for(int i=0;i< text.length(); i+=3)
        {
            outtext += encrypt(ptextchars[i],ptextchars[i+1],ptextchars[i+2]);
        }
        break;
        case 2:
        for(int i=0;i< text.length(); i+=3)
        {
            outtext += decrypt(ptextchars[i],ptextchars[i+1],ptextchars[i+2]);
        }
        break;
        default: System.out.println("Invalid Choice!");
    }
    System.out.println("Output: " + outtext);
}
```

```
        private static String encrypt(char a, char b, char c)
        {
                String ret = "";
                int x,y, z;
                int posa = (int)a - 65;
                int posb = (int)b - 65;
                int posc = (int)c - 65;
                x = posa * keymat[0][0] + posb * keymat[1][0] + posc * keymat[2][0];
                y = posa * keymat[0][1] + posb * keymat[1][1] + posc * keymat[2][1];
                z = posa * keymat[0][2] + posb * keymat[1][2] + posc * keymat[2][2];
                a = key.charAt(x%26);
                b = key.charAt(y%26);
                c = key.charAt(z%26);
                 ret = "" + a + b + c;
                return ret;
        }
        private static String decrypt(char a, char b, char c)
        {
                String ret = "";
                int x,y,z;
                int posa = (int)a - 65;
                int posb = (int)b - 65;
                int posc = (int)c - 65;
                x = posa * invkeymat[0][0]+ posb * invkeymat[1][0] + posc * invkeymat[2][0];
                y = posa * invkeymat[0][1]+ posb * invkeymat[1][1] + posc * invkeymat[2][1];
                z = posa * invkeymat[0][2]+ posb * invkeymat[1][2] + posc * invkeymat[2][2];
                a = key.charAt((x%26<0)?(26+x%26):(x%26));
                b = key.charAt((y%26<0)?(26+y%26):(y%26));
                c = key.charAt((z%26<0)?(26+z%26):(z%26));
                ret = "" + a + b + c;
                return ret;
        }
}
```

OUTPUT

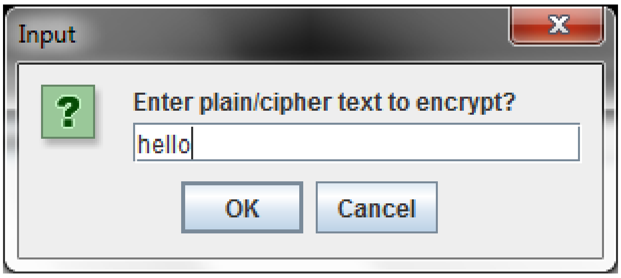
ENCRYPT



Input

Enter 1 to Encrypt and 2 to Decrypt!

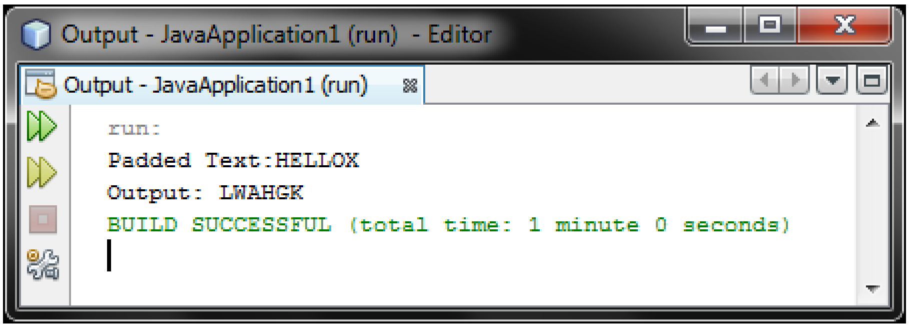
1



Input

Enter plain/cipher text to encrypt?

hello

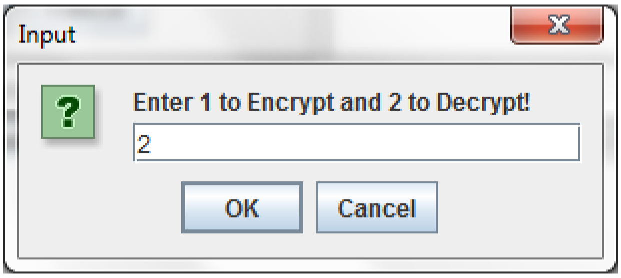


Output - JavaApplication1 (run) - Editor

Output - JavaApplication1 (run)

```
run:
Padded Text:HELLOX
Output: LWAHGK
BUILD SUCCESSFUL (total time: 1 minute 0 seconds)
```

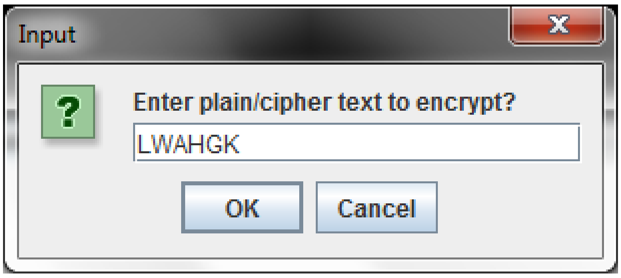
DECRYPT



Input

Enter 1 to Encrypt and 2 to Decrypt!

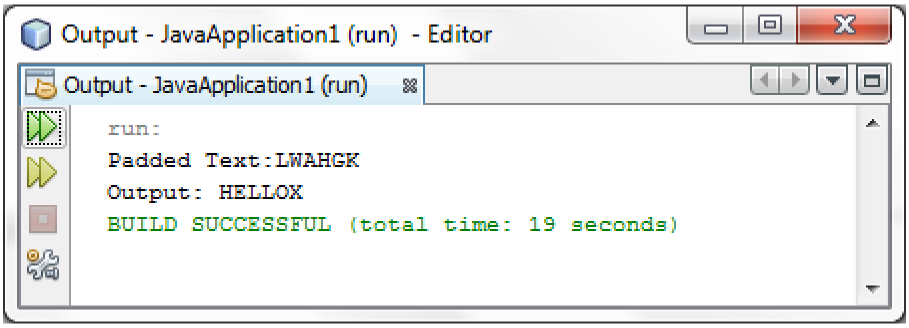
2



Input

Enter plain/cipher text to encrypt?

LWAHGK



Output - JavaApplication1 (run) - Editor

Output - JavaApplication1 (run)

```
run:
Padded Text:LWAHGK
Output: HELLOX
BUILD SUCCESSFUL (total time: 19 seconds)
```

RESULT

Thus java program to implement Hill Cipher was written, executed and output is verified successfully.

Ex.No: 1(iv) IMPLEMENTATION OF VIGENERE CIPHER

AIM

To write a Java Program for implementing Vigenere Cipher.

ALGORITHM

Step 1: Start the program.

Step 2: Define a class VC1, in that define encipher() to produce a cipher text.

Step 3: Define decipher() to reproduce the plain text.

Step 4: Define a shift() to shift the values.

Step 5: In main(), define the text and key values and call the encipher() to encrypt and decipher() to decrypt the encrypted text.

Step 6: Display the results.

Step 7: Stop the program.

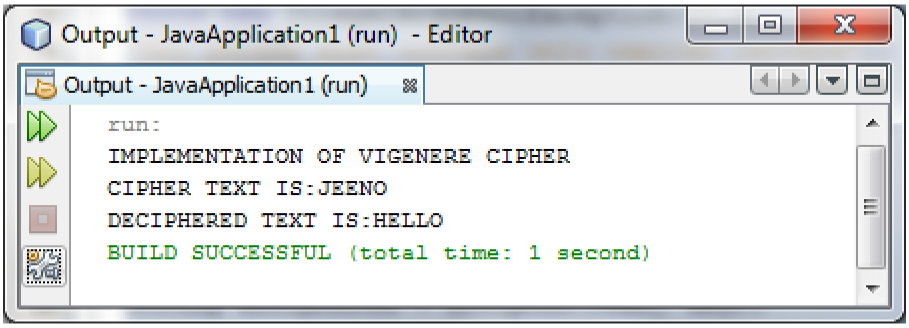
PROGRAM

```
package javaapplication1;
public class VC1
{
        public static String encipher(String s, String key)
        {
                StringBuilder builder = new StringBuilder();
                for(int i = 0; i < s.length(); i ++)
                {
                        if(s.charAt(i) < 65 || s.charAt(i) > 90)
                        { //ASCII character (capital letter)
                                throw new IllegalArgumentException("" +"Open text must contain
                                only capital letters");
                        }
                        //add shift modularly
                        char encyphered = s.charAt(i) + getShift(key, i) > 90 ? (char)((s.charAt(i) +
                        getShift(key, i)) - 26) : (char)(s.charAt(i) + getShift(key, i));
                        builder.append(encyphered);
                }
                return builder.toString();
```

```
}
public static String decipher(String s, String key)
{
    StringBuilder builder = new StringBuilder();
    for(int i = 0; i < s.length(); i ++)
    {
        if(s.charAt(i) < 65 || s.charAt(i) > 90)
        { //ASCII character (capital letter)
            throw new IllegalArgumentException("" +"Ciphertext must contain only capital letters");
        }
        //subtract shift modularly
        char decyphered = s.charAt(i) - getShift(key, i) < 65 ? (char)((s.charAt(i) - getShift(key, i)) + 26) : (char)(s.charAt(i) - getShift(key, i));
        builder.append(decyphered);
    }
    return builder.toString();
}
private static int getShift(String key, int i)
{
    if(key.charAt(i % key.length()) < 65 || key.charAt(i % key.length()) > 90)
    {
        throw new IllegalArgumentException("" +"Key phrase must contain only capital letters");
    }
    return ((int)key.charAt(i % key.length())) - 65;
}
public static void main(String[] args)
{
    String text = "HELLO";
    String key = "CAT";
    String enciphered = encipher(text, key);
    System.out.println("IMPLEMENTATION OF VIGENERE CIPHER");
    System.out.println("CIPHER TEXT IS:"+enciphered);
    System.out.println("DECIPHERED TEXT IS:"+decipher(enciphered, key));
```

```
        }
}
```

OUTPUT


Output - JavaApplication1 (run) - Editor
Output - JavaApplication1 (run)

```
run:
IMPLEMENTATION OF VIGENERE CIPHER
CIPHER TEXT IS:JEENO
DECIPHERED TEXT IS:HELLO
BUILD SUCCESSFUL (total time: 1 second)
```

RESULT

Thus java program to implement Vigenere Cipher was written, executed and output is verified successfully.

Ex.No: 2 **IMPLEMENTATION OF RAIL FENCE –**

ROW & COLUMN TRANSFORMATION

AIM

To write a Java Program for implementing Rail Fence – Row & Column Transformation.

ALGORITHM

Step 1: Start the program.

Step 2: Define a class railfencebasic, in that define Encryption() to produce cipher text.

Step 3: Define Decryption() to produce decipher text.

Step 4: Define a class railfencecipher, in that define main() and input the text to cipher and decipher it.

Step 5: Display the results.

Step 6: Stop the program.

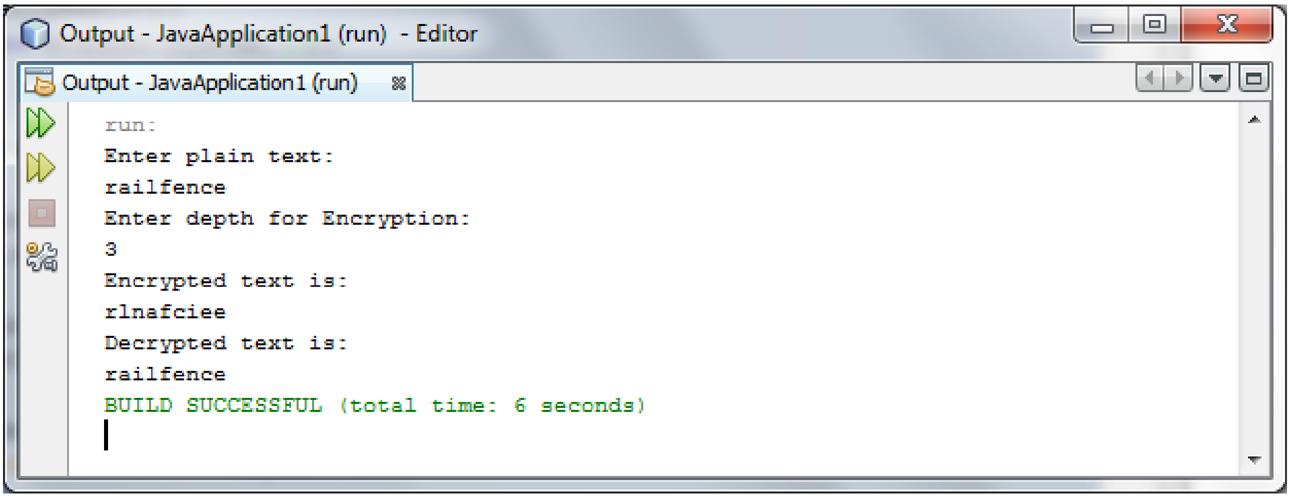
PROGRAM

```
package javaapplication1;
import java.util.*;
class RailFenceBasic
{
        int depth;
        String Encryption(String plainText,int depth)throws Exception
        {
                int r=depth,len=plainText.length();
                int c=len/depth;
                char mat[][]=new char[r][c];
                int k=0;
                String cipherText="";
                for(int i=0;i< c;i++)
                {
                        for(int j=0;j< r;j++)
                        {
                                if(k!=len)
                                        mat[j][i]=plainText.charAt(k++);
```

```
            else
                mat[j][i]='X';
        }
    }
    for(int i=0;i< r;i++)
    {
        for(int j=0;j< c;j++)
        {
            cipherText+=mat[i][j];
        }
    }
    return cipherText;
}
String Decryption(String cipherText,int depth)throws Exception
{
    int r=depth,len=cipherText.length();
    int c=len/depth;
    char mat[][]=new char[r][c];
    int k=0;
    String plainText="";
    for(int i=0;i< r;i++)
    {
        for(int j=0;j< c;j++)
        {
            mat[i][j]=cipherText.charAt(k++);
        }
    }
    for(int i=0;i< c;i++)
    {
        for(int j=0;j< r;j++)
        {
            plainText+=mat[j][i];
        }
    }
    return plainText; }
```

```
}
 class railfencecipher
{
	public static void main(String args[])throws Exception
	{
		RailFenceBasic rf=new RailFenceBasic();
		Scanner scn=new Scanner(System.in);
		int depth;
		String plainText,cipherText,decryptedText;
		System.out.println("Enter plain text:");
		plainText=scn.nextLine();
		System.out.println("Enter depth for Encryption:");
		depth=scn.nextInt();
		cipherText=rf.Encryption(plainText,depth);
		System.out.println("Encrypted text is:\n"+cipherText);
		decryptedText=rf.Decryption(cipherText, depth);
		System.out.println("Decrypted text is:\n"+decryptedText);
	}
}
```

OUTPUT


Output - JavaApplication1 (run) - Editor
Output - JavaApplication1 (run)

```
run:
Enter plain text:
railfence
Enter depth for Encryption:
3
Encrypted text is:
rlnafciee
Decrypted text is:
railfence
BUILD SUCCESSFUL (total time: 6 seconds)
```

RESULT

Thus java program to implement Rail Fence – Row & Column Transformation was written, executed and output is verified successfully.

Ex.No: 3 IMPLEMENTATION OF DES

AIM

To write a Java Program for implementing DES.

ALGORITHM

Step 1: Start the program.

Step 2: Define a class DES, in that define a constructor to display the encrypted and decrypted message.

Step 3: Define generateSymmetricKey(), to generate the key.

Step 4: Define encrypt() to encrypt the plain text.

Step 5: Define decrypt() to decrypt the ciphered text.

Step 6: Define main() to declare object for the class.

Step 7: Stop the program.

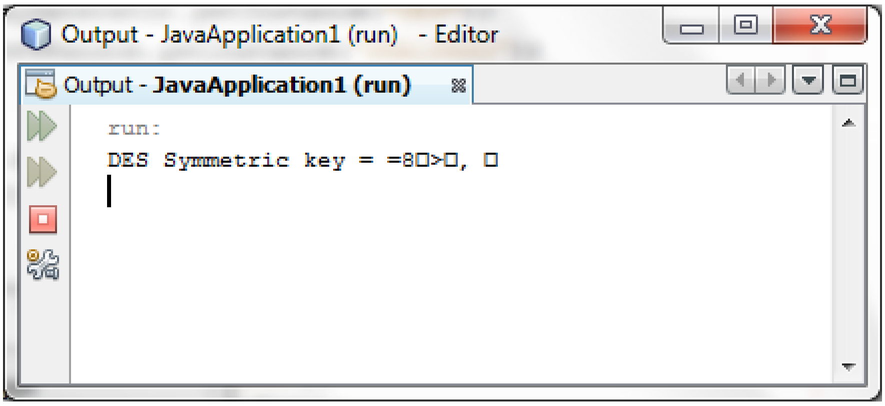
PROGRAM

```
package javaapplication1;
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
                {
```

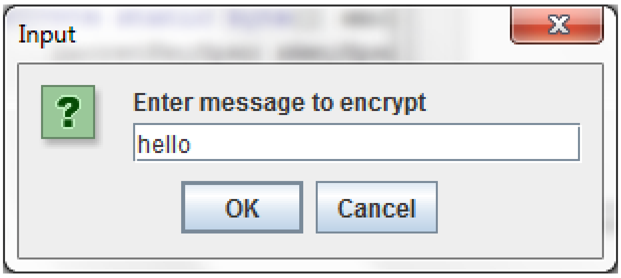
```
            generateSymmetricKey();
            inputMessage=JOptionPane.showInputDialog(null,"Enter message to
            encrypt");
            byte[] ibyte = inputMessage.getBytes();
            byte[] ebyte=encrypt(raw, ibyte);
            String encryptedData = new String(ebyte);
            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

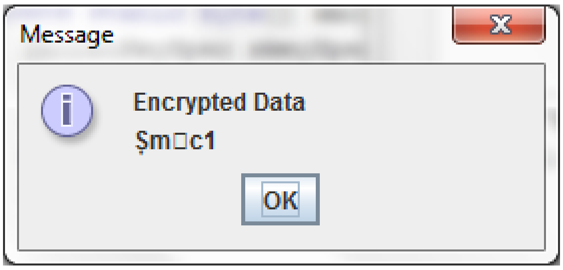

Output - JavaApplication1 (run) - Editor
Output - JavaApplication1 (run)

```
run:
DES Symmetric key = =8□>□, □
```



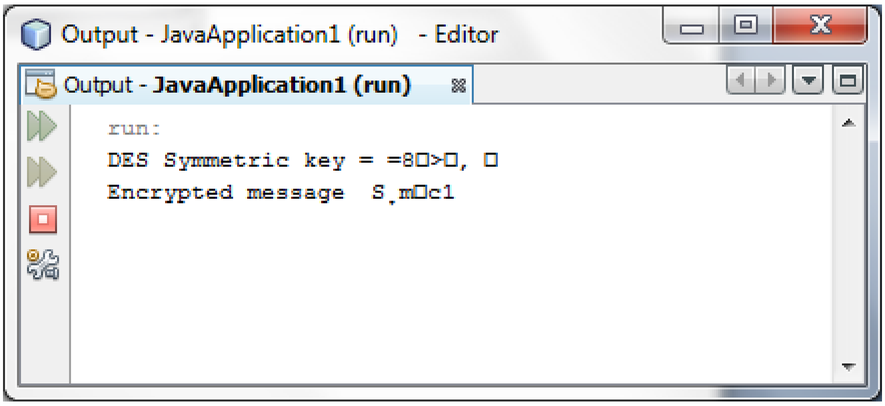
Enter message to encrypt

hello

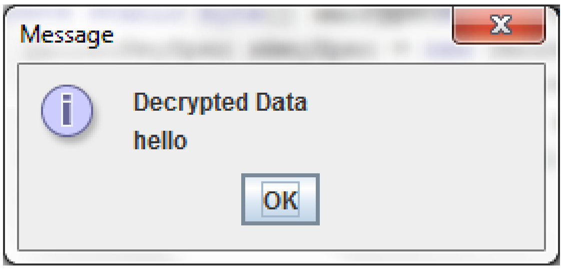


Encrypted Data

Şm□c1

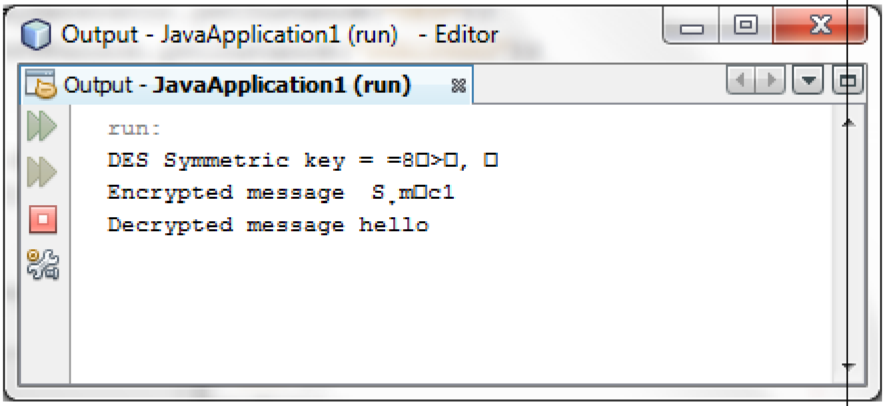

Output - JavaApplication1 (run) - Editor
Output - JavaApplication1 (run)

```
run:
DES Symmetric key = =8□>□, □
Encrypted message  Şm□c1
```



Decrypted Data

hello


Output - JavaApplication1 (run) - Editor
Output - JavaApplication1 (run)

```
run:
DES Symmetric key = =8□>□, □
Encrypted message  Şm□c1
Decrypted message hello
```

RESULT

Thus java program to implement DES was written, executed and output is verified successfully.

Ex.No:4 IMPLMENTATION OF AES

AIM

To implement AES using java.

ALGORITHM

Step 1: Start the program.

Step 2: Define a class AES, in that assign values to plaintext, IV and key variables.

Step 3: Define main() to display the encrypted and decrypted text of plain text.

Step 4: Define encrypt() to generated cipher text and decrypt() to generate plain text.

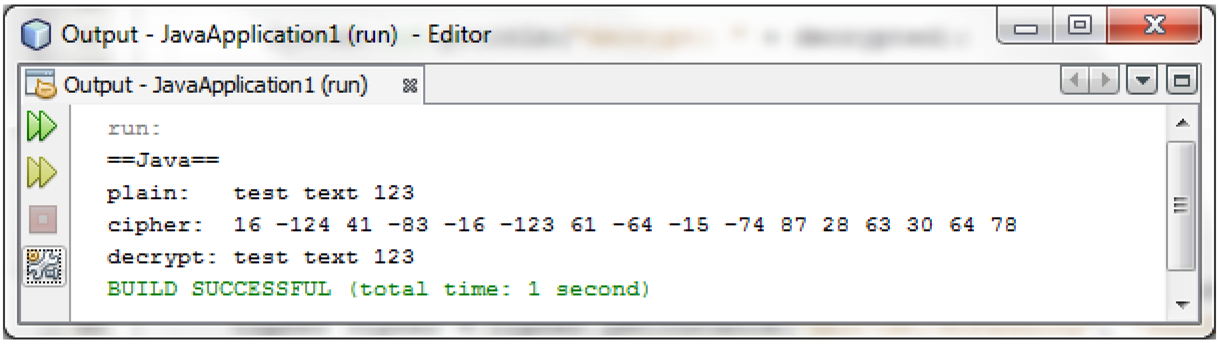
Step 5: Stop the program.

PROGRAM

```
package javaapplication1;
import java.security.MessageDigest;
import java.util.Arrays;
import javax.crypto.KeyGenerator;
import javax.crypto.SecretKey;
import javax.crypto.spec.SecretKeySpec;
import javax.crypto.spec.IvParameterSpec;

import javax.crypto.Cipher;
import javax.crypto.spec.IvParameterSpec;
import javax.crypto.spec.SecretKeySpec;

public class AES
{
        static String IV = "AAAAAAAAAAAAAAAA";
        static String plaintext = "test text 123\0\0\0"; /*Note null padding*/
        static String encryptionKey = "0123456789abcdef";
        public static void main(String [] args)
        {
                try
                {
                        System.out.println("==Java==");
```

```
            System.out.println("plain:   " + plaintext);
            byte[] cipher = encrypt(plaintext, encryptionKey);
            System.out.print("cipher:  ");
            for (int i=0; i<cipher.length; i++)
                System.out.print(new Integer(cipher[i])+" ");
            System.out.println("");
            String decrypted = decrypt(cipher, encryptionKey);
            System.out.println("decrypt: " + decrypted);
        }
        catch (Exception e)
        {
            e.printStackTrace();
        }
    }
    public static byte[] encrypt(String plainText, String encryptionKey) throws Exception
    {
        Cipher cipher = Cipher.getInstance("AES/CBC/NoPadding", "SunJCE");
        SecretKeySpec key = new SecretKeySpec(encryptionKey.getBytes("UTF-8"),
        "AES");
        cipher.init(Cipher.ENCRYPT_MODE,key,new
        IvParameterSpec(IV.getBytes("UTF-8")));
        return cipher.doFinal(plainText.getBytes("UTF-8"));
    }
    public static String decrypt(byte[] cipherText, String encryptionKey) throws Exception
    {
        Cipher cipher = Cipher.getInstance("AES/CBC/NoPadding", "SunJCE");
        SecretKeySpec key = new SecretKeySpec(encryptionKey.getBytes("UTF-8"),
        "AES");
        cipher.init(Cipher.DECRYPT_MODE,key,new
        IvParameterSpec(IV.getBytes("UTF-8")));
        return new String(cipher.doFinal(cipherText),"UTF-8");
    }
}
```

OUTPUT


Output - JavaApplication1 (run) - Editor
Output - JavaApplication1 (run)

```
run:
==Java==
plain:   test text 123
cipher:  16 -124 41 -83 -16 -123 61 -64 -15 -74 87 28 63 30 64 78
decrypt: test text 123
BUILD SUCCESSFUL (total time: 1 second)
```

RESULT

Thus AES was implemented using java and output is verified successfully.

Ex.No: 5 **IMPLEMENTATION OF RSA**

AIM

To write a Java Program for implementing RSA.

ALGORITHM

```
Step 1: Start the program.
Step 2: Define a default constructor RSA to compare 2 prime numbers.
Step 3: Define a parameterized constructor to assign values.
Step 4: Define bytetostring() to convert byte to string.
Step 5: Define encrypt() to encrypt the text and decrypt() to reproduce the plain text.
Step 6: Define main() to call the encrypt() and decrypt() to perform respective operations and display the results.
Step 7: Stop the program.
```

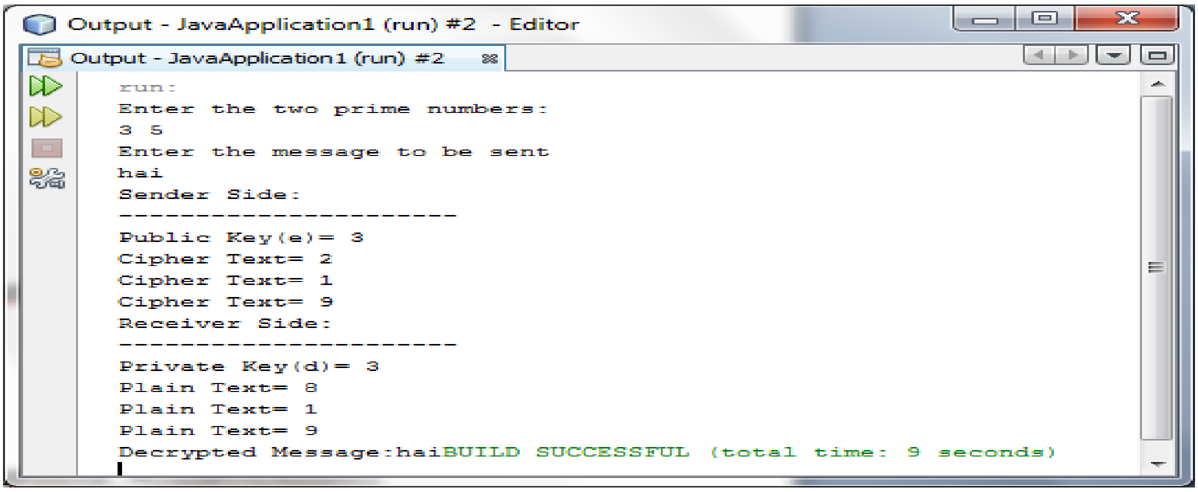
PROGRAM

```
package javaapplication1;
import java.io.DataInputStream;
import java.io.IOException;
import java.math.BigInteger;
import java.util.Random;
public class RSA
{
        private BigInteger p;
        private BigInteger q;
        private BigInteger N;
         private BigInteger phi;
        private BigInteger e;
        private BigInteger d;
         private int        bitlength = 1024;
         private Random     r;
        public RSA()
        {
                r = new Random();
                p = BigInteger.probablePrime(bitlength, r);
```

```
    q = BigInteger.probablePrime(bitlength, r);
    N = p.multiply(q);
    phi = p.subtract(BigInteger.ONE).multiply(q.subtract(BigInteger.ONE));
    e = BigInteger.probablePrime(bitlength / 2, r);
    while (phi.gcd(e).compareTo(BigInteger.ONE) > 0 && e.compareTo(phi) < 0)
    {
        e.add(BigInteger.ONE);
    }
    d = e.modInverse(phi);
 }
public RSA(BigInteger e, BigInteger d, BigInteger N)
{
    this.e = e;
    this.d = d;
    this.N = N;
}
@SuppressWarnings("deprecation")
public static void main(String[] args) throws IOException
{
    RSA rsa = new RSA();
    DataInputStream in = new DataInputStream(System.in);
    String teststring;
    System.out.println("Enter the plain text:");
    teststring = in.readLine();
    System.out.println("Encrypting String: " + teststring);
    System.out.println("String in Bytes: "
    + bytesToString(teststring.getBytes()));
    // encrypt
    byte[] encrypted = rsa.encrypt(teststring.getBytes());
    // decrypt
    byte[] decrypted = rsa.decrypt(encrypted);
    System.out.println("Decrypting Bytes: " + bytesToString(decrypted));
    System.out.println("Decrypted String: " + new String(decrypted));
}
```

```
        private static String bytesToString(byte[] encrypted)
        {
                String test = "";
                for (byte b : encrypted)
                {
                        test += Byte.toString(b);
                }
                return test;
        }
    // Encrypt message
        public byte[] encrypt(byte[] message)
        {
                return (new BigInteger(message)).modPow(e, N).toByteArray();
        }
    // Decrypt message
        public byte[] decrypt(byte[] message)
        {
                return (new BigInteger(message)).modPow(d, N).toByteArray();
        }
}
```

OUTPUT


Output - JavaApplication1 (run) #2 - Editor
Output - JavaApplication1 (run) #2

```
run:
Enter the two prime numbers:
3 5
Enter the message to be sent
hai
Sender Side:
----------------------
Public Key(e)= 3
Cipher Text= 2
Cipher Text= 1
Cipher Text= 9
Receiver Side:
----------------------
Private Key(d)= 3
Plain Text= 8
Plain Text= 1
Plain Text= 9
Decrypted Message:haiBUILD SUCCESSFUL (total time: 9 seconds)
```

RESULT

Thus java program to implement RSA was written, executed and output is verified successfully.

Ex.No: 6 IMPLEMENTATION OF DIFFIEE - HELLMAN

AIM

To write a Java Program for implementing Diffiee - Hellman.

ALGORITHM

Step 1: Start the program.

Step 2: Define class DiffeHellmanBigInt, in that main() get the details and pass the secret key.

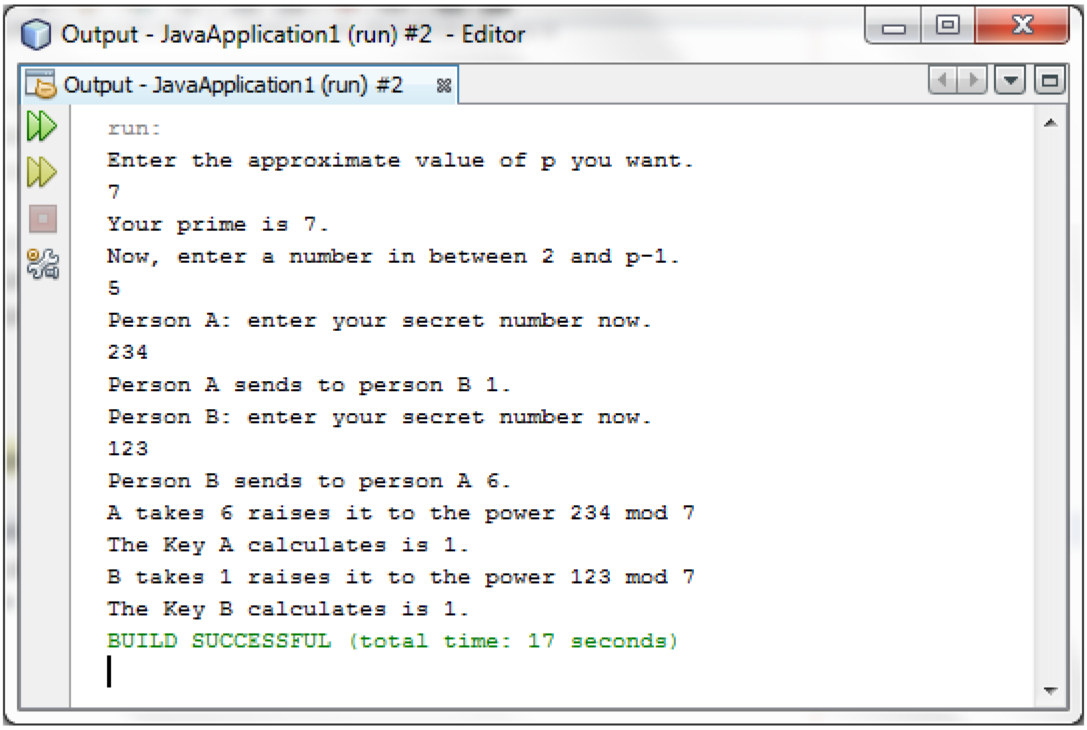
Step 3: Calculate the keys and display it.

Step 4: Stop the program.

PROGRAM

```
package javaapplication1;
import java.util.*;
import java.math.BigInteger;
public class DiffeHellmanBigInt
{
        final static BigInteger one = new BigInteger("1");
        public static void main(String args[])
        {
                Scanner stdin = new Scanner(System.in);
                BigInteger p;
                // Get a start spot to pick a prime from the user.
                System.out.println("Enter the approximate value of p you want.");
                String ans = stdin.next();
                p = getNextPrime(ans);
                System.out.println("Your prime is "+p+".");
                // Get the base for exponentiation from the user.
                System.out.println("Now, enter a number in between 2 and p-1.");
                BigInteger g = new BigInteger(stdin.next());
                // Get A's secret number.
                System.out.println("Person A: enter your secret number now.");
                BigInteger a = new BigInteger(stdin.next());
                // Make A's calculation.
                BigInteger resulta = g.modPow(a,p);
                // This is the value that will get sent from A to B.
                // This value does NOT compromise the value of a easily.
                System.out.println("Person A sends to person B "+resulta+".");
                // Get B's secret number.
                System.out.println("Person B: enter your secret number now.");
                BigInteger b = new BigInteger(stdin.next());
                // Make B's calculation.
                BigInteger resultb = g.modPow(b,p);
                // This is the value that will get sent from B to A.
                // This value does NOT compromise the value of b easily.
                System.out.println("Person B sends to person A "+resultb+".");
                // Once A and B receive their values, they make their new calculations.
                // This involved getting their new numbers and raising them to the
                // same power as before, their secret number.
                BigInteger KeyACalculates = resultb.modPow(a,p);
                BigInteger KeyBCalculates = resulta.modPow(b,p);
                // Print out the Key A calculates.
                System.out.println("A takes "+resultb+" raises it to the power "+a+" mod "+p);
                System.out.println("The Key A calculates is "+KeyACalculates+".");
                // Print out the Key B calculates.
                System.out.println("B takes "+resulta+" raises it to the power "+b+" mod "+p);
                System.out.println("The Key B calculates is "+KeyBCalculates+".");
        }
        public static BigInteger getNextPrime(String ans)
        {
                BigInteger test = new BigInteger(ans);
                while (!test.isProbablePrime(99))
                    test = test.add(one);
                return test;
        }

}
```

OUTPUT


Output - JavaApplication1 (run) #2 - Editor
Output - JavaApplication1 (run) #2

```
run:
Enter the approximate value of p you want.
7
Your prime is 7.
Now, enter a number in between 2 and p-1.
5
Person A: enter your secret number now.
234
Person A sends to person B 1.
Person B: enter your secret number now.
123
Person B sends to person A 6.
A takes 6 raises it to the power 234 mod 7
The Key A calculates is 1.
B takes 1 raises it to the power 123 mod 7
The Key B calculates is 1.
BUILD SUCCESSFUL (total time: 17 seconds)
```

RESULT

Thus java program to implement Diffiee Hellman was written, executed and output is verified successfully.

Ex.No: 7 IMPLEMENTATION OF SHA-1

AIM

To write a Java Program for implementing SHA-1.

ALGORITHM

Step 1: Start the program.

Step 2: Define a class HashTextTest, in that main() call sha1() to display secured hash value.

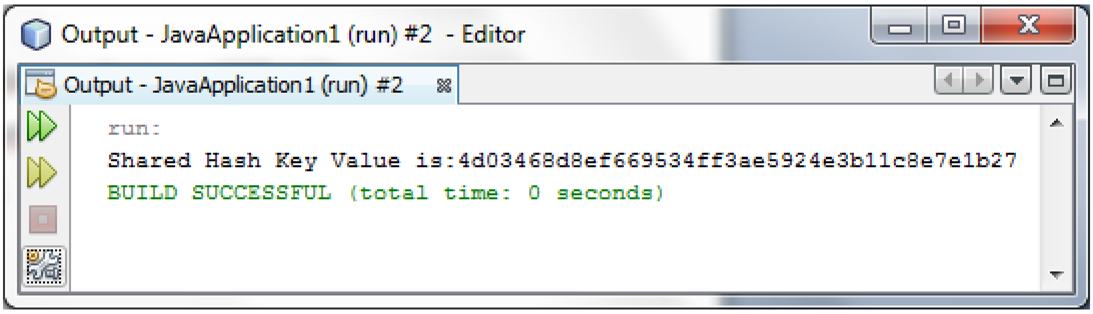
Step 3: Define sha1(), in that define the instances and generate the hash value.

Step 4: Stop the program.

PROGRAM

```
package javaapplication1;
import java.security.MessageDigest;
import java.security.NoSuchAlgorithmException;
public class HashTextTest
{
        public static void main(String[] args) throws NoSuchAlgorithmException
        {
                System.out.println("Shared Hash Key Value is:"+sha1("shared Hashing"));
        }
        static String sha1(String input) throws NoSuchAlgorithmException
        {
                MessageDigest mDigest = MessageDigest.getInstance("SHA1");
                byte[] result = mDigest.digest(input.getBytes());
                StringBuffer sb = new StringBuffer();
                for (int i = 0; i < result.length; i++)
                {
                        sb.append(Integer.toString((result[i] & 0xff) + 0x100, 16).substring(1));
                }
                return sb.toString();
        }
}
```

OUTPUT



Output - JavaApplication1 (run) #2 - Editor
Output - JavaApplication1 (run) #2

```
run:
Shared Hash Key Value is:4d03468d8ef669534ff3ae5924e3b11c8e7e1b27
BUILD SUCCESSFUL (total time: 0 seconds)
```

RESULT

Thus java program to implement SHA was written, executed and output is verified successfully.

Ex.No: 8 **IMPLEMENTATION OF SIGNATURE SCHEME**

DIGITAL SIGNATURE STANDARD

AIM

To implement Signature Scheme of Digital Signature Standard using java.

ALGORITHM

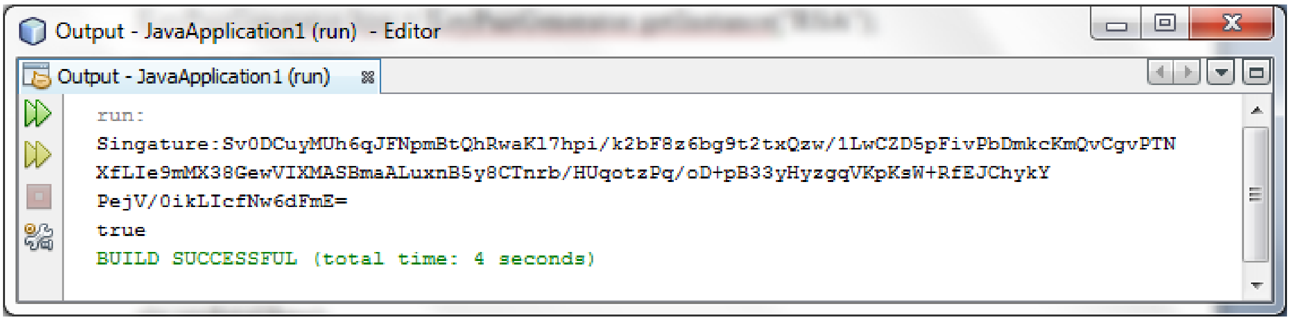
```
Step 1: Start the program.
Step 2: Define a class DSS, in that define a main().
Step 3: Create instance for KeypairGenerator class.
Step 4: Generate Key Pair using KeyPair Class.
Step 5: Send the data to encrypt and sign.
Step 6: Sign the data using Signature class.
Step 7: Display the signature and verification status.
Step 8: Stop the program.
```

PROGRAM

```
package javaapplication1;
import java.security.KeyPair;
import java.security.KeyPairGenerator;
import java.security.Signature;
import sun.misc.BASE64Encoder;
public class DSS
{
	public static void main(String[] args) throws Exception
	{
		KeyPairGenerator kpg = KeyPairGenerator.getInstance("RSA");
		kpg.initialize(1024);
		KeyPair keyPair = kpg.genKeyPair();
		byte[] data = "test".getBytes("UTF8");
		Signature sig = Signature.getInstance("MD5WithRSA");
		sig.initSign(keyPair.getPrivate());
		sig.update(data);
		byte[] signatureBytes = sig.sign();
		System.out.println("Singature:" + new BASE64Encoder().encode(signatureBytes));
```

```
                    sig.initVerify(keyPair.getPublic());
                    sig.update(data);
                    System.out.println(sig.verify(signatureBytes));
          }
}
```

OUTPUT


Output - JavaApplication1 (run) - Editor
Output - JavaApplication1 (run)

```
run:
Singature:Sv0DCuyMUh6qJFNpmBtQhRwaKl7hpi/k2bF8z6bg9t2txQzw/1LwCZD5pFivPbDmkcKmQvCgvPTN
XfLIe9mMX38GewVIXMASBmaALuxnB5y8CTnrb/HUqotzPq/oD+pB33yHyzgqVKpKsW+RfEJChykY
PejV/0ikLIcfNw6dFmE=
true
BUILD SUCCESSFUL (total time: 4 seconds)
```

RESULT

Thus java program to implement Signature Scheme of Digital Signature Standard was written, executed and output is verified successfully.

Ex.No: 9 DEMONSTRATE INTRUSION DETECTION SYSTEM (IDS) USING SNORT

AIM

To demonstrate intrusion detection system (ids) using snort.

PROCEDURE

SNORT can be configured to run in three modes:

1. Sniffer mode
2. Packet Logger mode
3. Network Intrusion Detection System mode

Sniffer mode→snort –v Print out the TCP/IP packets header on the screen

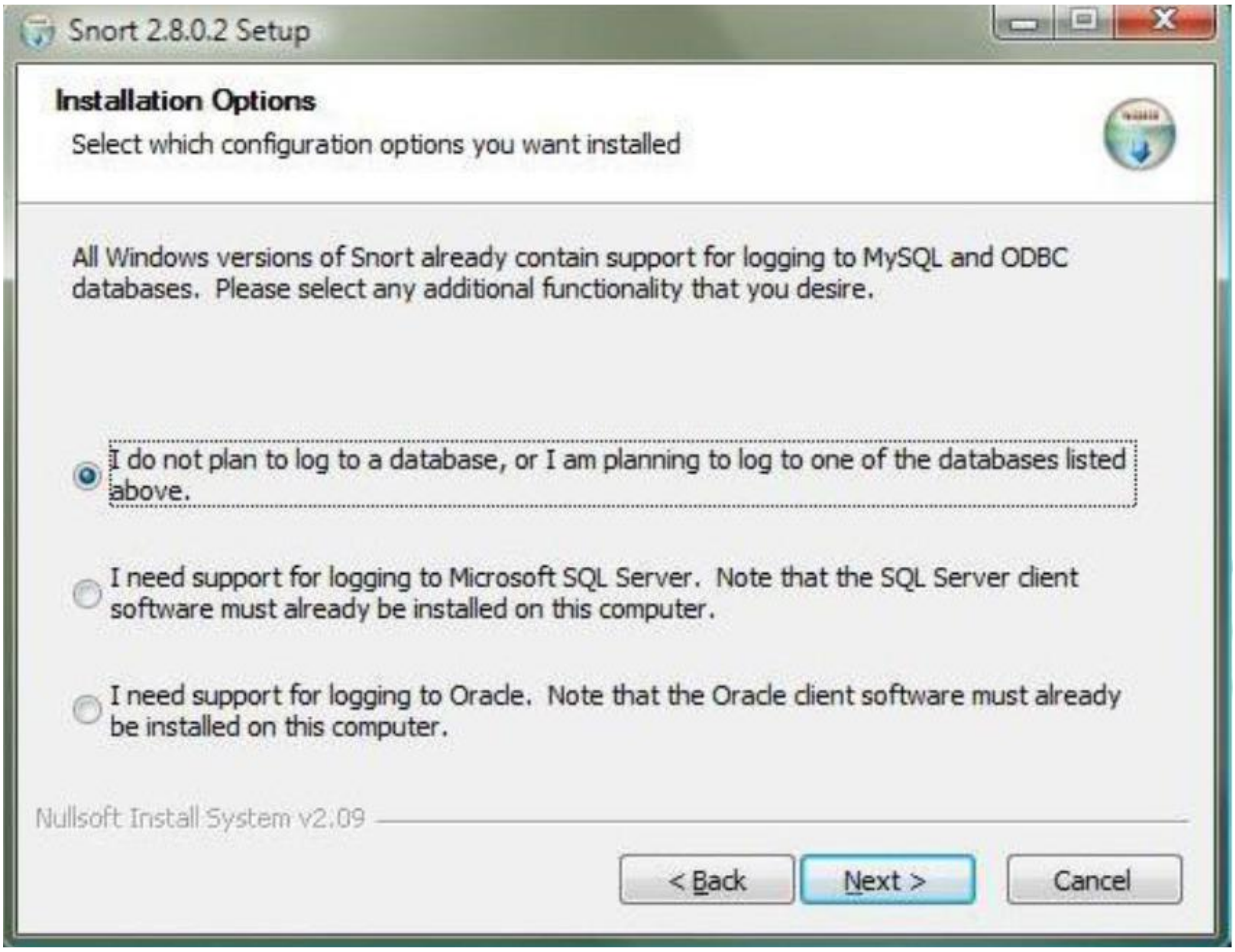
Snort –vd show the TCP/IP ICMP header with application data in transit.

Packet Logger mode→snort –dev –l c:\log [create this directory in the C drive] and snort will automatically know to go into packet logger mode, it collects every packet it sees and places it in log directory. snort –dev –l c:\log –h ipaddress/24 This rule tells snort that you want to print out the data link and TCP/IP headers as well as application data into the log directory. snort –l c:\log –b This is binary mode logs everything into a single file.

Network Intrusion Detection System mode→snort –d c:\log –h ipaddress/24 –c snort.conf - This is a configuration file applies rule to each packet to decide it an action based upon the rule type in the file. Snort –d –h ipaddress/24 –l c:\log –c snort.conf - This will configure snort to run in its most basic NIDS form, logging packets that trigger rules specifies in the snort.conf.

Step 1: Download SNORT from snort.org

Step 2: Install snort with or without database support.



Installation Options

Select which configuration options you want installed

All Windows versions of Snort already contain support for logging to MySQL and ODBC databases. Please select any additional functionality that you desire.

● I do not plan to log to a database, or I am planning to log to one of the databases listed above.

○ I need support for logging to Microsoft SQL Server. Note that the SQL Server client software must already be installed on this computer.

○ I need support for logging to Oracle. Note that the Oracle client software must already be installed on this computer.

Nullsoft Install System v2.09

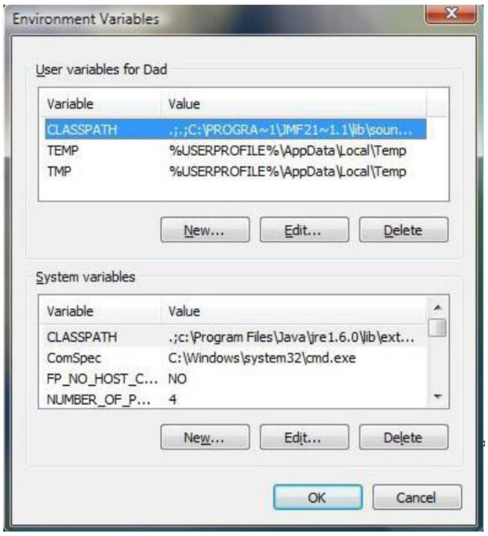
Step 3: Select all the components and Click Next.

Step 4: Install and Close.

Step 5: Skip the WinPcap driver installation

Step 6; Add the path variable in windows environment variable by selecting new classpath.

Step 7: Create a path variable and point it at snort.exe variable name→path and variable value→c:\snort\bin.



User variables for Dad

Variable	Value
CLASSPATH	.;.;C:\PROGRA~1\JMF21~1.1\lib\soun...
TEMP	%USERPROFILE%\AppData\Local\Temp
TMP	%USERPROFILE%\AppData\Local\Temp

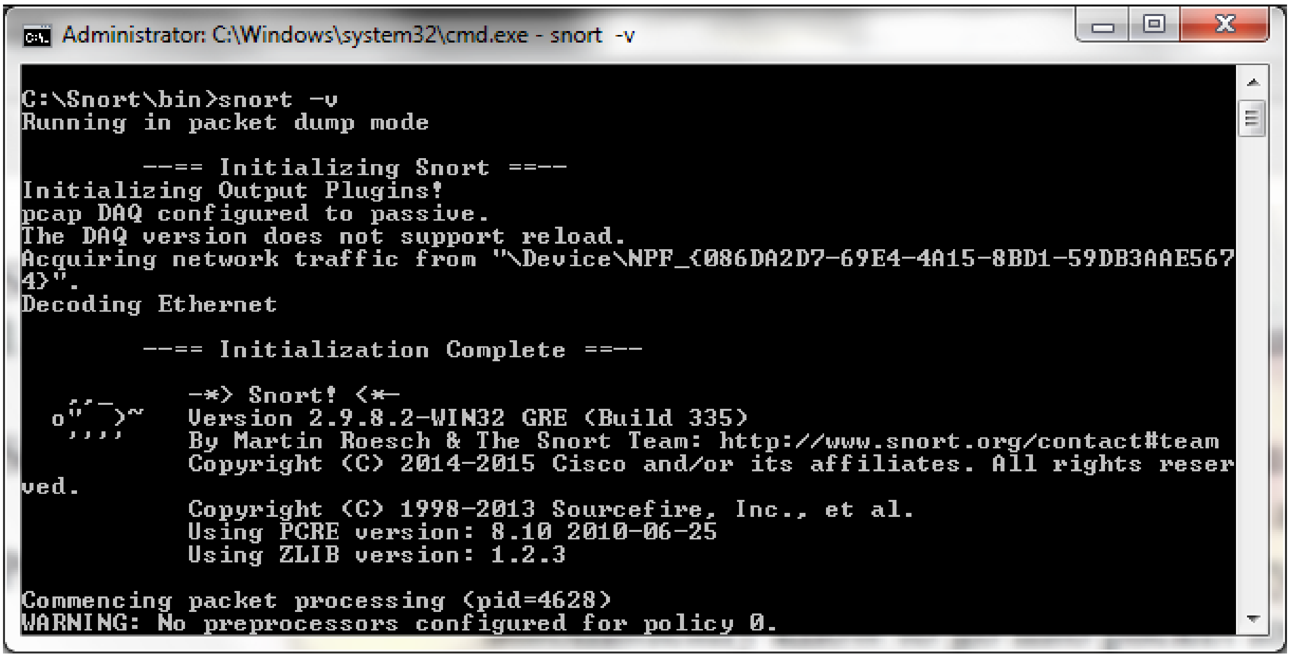
System variables

Variable	Value
CLASSPATH	.;c:\Program Files\Java\jre1.6.0\lib\ext...
ComSpec	C:\Windows\system32\cmd.exe
FP_NO_HOST_C...	NO
NUMBER_OF_P...	4

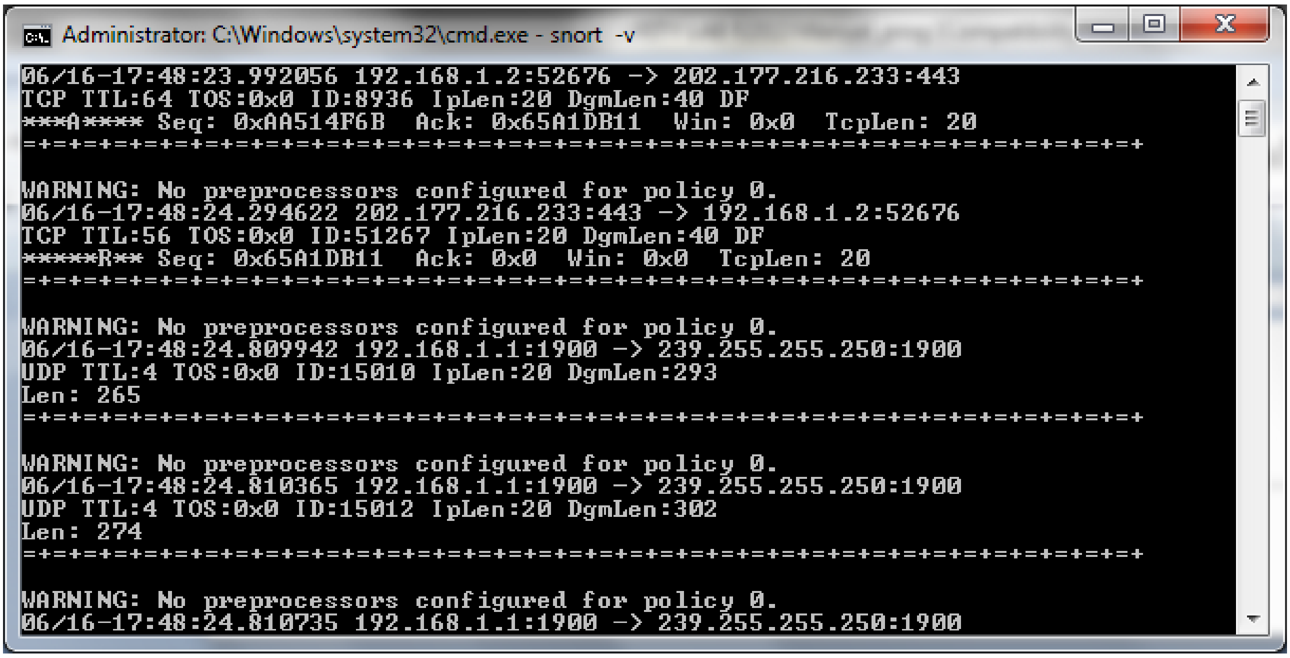
Step 8: Click OK button and then close all dialog boxes.

Step 9: Open command prompt and type the following commands:

C:\Snort\bin>Snort - v

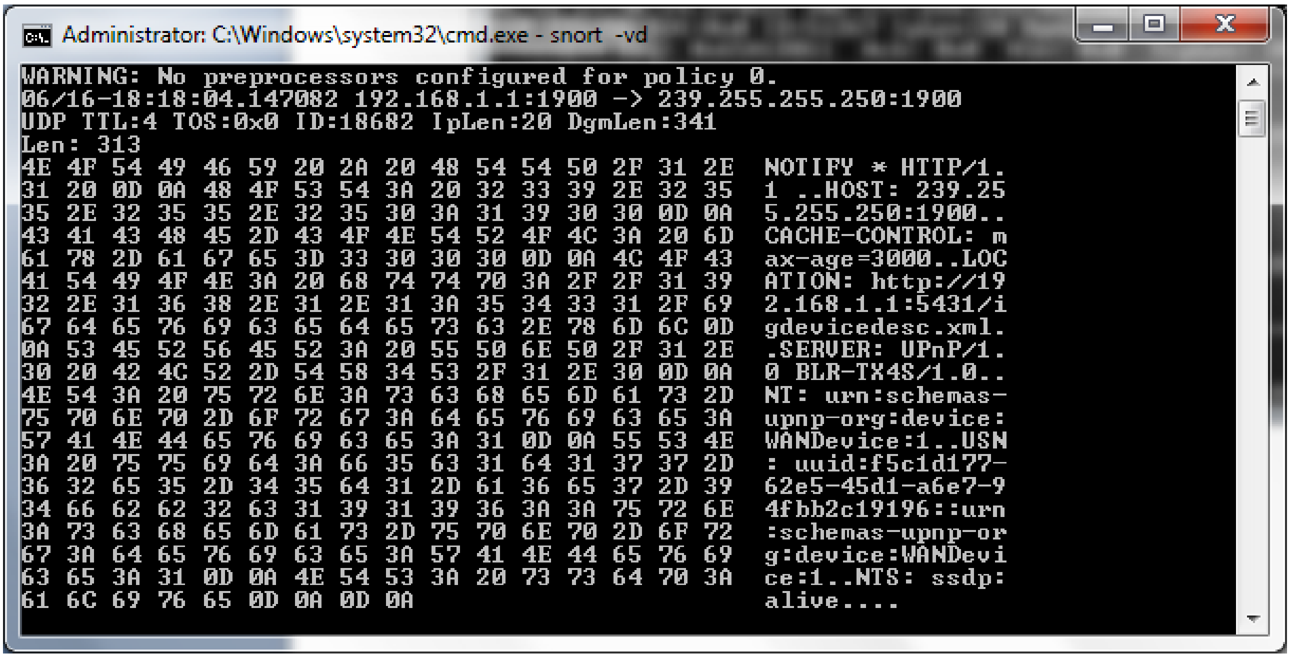


```
C:\Snort\bin>snort -v
Running in packet dump mode

        --== Initializing Snort ==--
Initializing Output Plugins!
pcap DAQ configured to passive.
The DAQ version does not support reload.
Acquiring network traffic from "\Device\NPF_{086DA2D7-69E4-4A15-8BD1-59DB3AAE567
4}".
Decoding Ethernet

        --== Initialization Complete ==--

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.9.8.2-WIN32 GRE (Build 335)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/contact#team
           Copyright (C) 2014-2015 Cisco and/or its affiliates. All rights reser
ved.
           Copyright (C) 1998-2013 Sourcefire, Inc., et al.
           Using PCRE version: 8.10 2010-06-25
           Using ZLIB version: 1.2.3

Commencing packet processing (pid=4628)
WARNING: No preprocessors configured for policy 0.
```

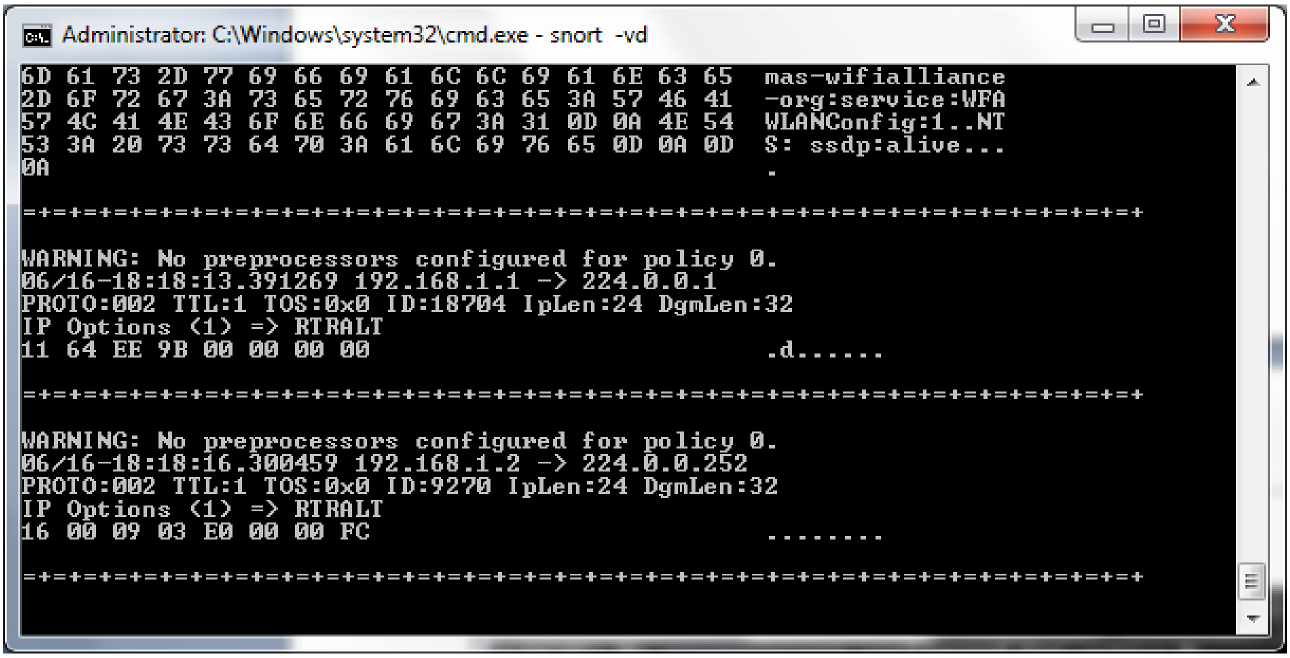


```
06/16-17:48:23.992056 192.168.1.2:52676 -> 202.177.216.233:443
TCP TTL:64 TOS:0x0 ID:8936 IpLen:20 DgmLen:40 DF
***A**** Seq: 0xAA514F6B  Ack: 0x65A1DB11  Win: 0x0  TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
06/16-17:48:24.294622 202.177.216.233:443 -> 192.168.1.2:52676
TCP TTL:56 TOS:0x0 ID:51267 IpLen:20 DgmLen:40 DF
*****R** Seq: 0x65A1DB11  Ack: 0x0  Win: 0x0  TcpLen: 20
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
06/16-17:48:24.809942 192.168.1.1:1900 -> 239.255.255.250:1900
UDP TTL:4 TOS:0x0 ID:15010 IpLen:20 DgmLen:293
Len: 265
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
06/16-17:48:24.810365 192.168.1.1:1900 -> 239.255.255.250:1900
UDP TTL:4 TOS:0x0 ID:15012 IpLen:20 DgmLen:302
Len: 274
=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
06/16-17:48:24.810735 192.168.1.1:1900 -> 239.255.255.250:1900
```

C:\Snort\bin>Snort – vd



```
WARNING: No preprocessors configured for policy 0.
06/16-18:18:04.147082 192.168.1.1:1900 -> 239.255.255.250:1900
UDP TTL:4 TOS:0x0 ID:18682 IpLen:20 DgmLen:341
Len: 313
4E 4F 54 49 46 59 20 2A 20 48 54 54 50 2F 31 2E  NOTIFY * HTTP/1.
31 20 0D 0A 48 4F 53 54 3A 20 32 33 39 2E 32 35  1 ..HOST: 239.25
35 2E 32 35 35 2E 32 35 30 3A 31 39 30 30 0D 0A  5.255.250:1900..
43 41 43 48 45 2D 43 4F 4E 54 52 4F 4C 3A 20 6D  CACHE-CONTROL: m
61 78 2D 61 67 65 3D 33 30 30 30 0D 0A 4C 4F 43  ax-age=3000..LOC
41 54 49 4F 4E 3A 20 68 74 74 70 3A 2F 2F 31 39  ATION: http://19
32 2E 31 36 38 2E 31 2E 31 3A 35 34 33 31 2F 69  2.168.1.1:5431/i
67 64 65 76 69 63 65 64 65 73 63 2E 78 6D 6C 0D  gdevicedesc.xml.
0A 53 45 52 56 45 52 3A 20 55 50 6E 50 2F 31 2E  .SERVER: UPnP/1.
30 20 42 4C 52 2D 54 58 34 53 2F 31 2E 30 0D 0A  0 BLR-TX4S/1.0..
4E 54 3A 20 75 72 6E 3A 73 63 68 65 6D 61 73 2D  NT: urn:schemas-
75 70 6E 70 2D 6F 72 67 3A 64 65 76 69 63 65 3A  upnp-org:device:
57 41 4E 44 65 76 69 63 65 3A 31 0D 0A 55 53 4E  WANDevice:1..USN
3A 20 75 75 69 64 3A 66 35 63 31 64 31 37 37 2D  : uuid:f5c1d177-
36 32 65 35 2D 34 35 64 31 2D 61 36 65 37 2D 39  62e5-45d1-a6e7-9
34 66 62 62 32 63 31 39 31 39 36 3A 3A 75 72 6E  4fbb2c19196::urn
3A 73 63 68 65 6D 61 73 2D 75 70 6E 70 2D 6F 72  :schemas-upnp-or
67 3A 64 65 76 69 63 65 3A 57 41 4E 44 65 76 69  g:device:WANDevi
63 65 3A 31 0D 0A 4E 54 53 3A 20 73 73 64 70 3A  ce:1..NTS: ssdp:
61 6C 69 76 65 0D 0A 0D 0A                       alive....
```



```
6D 61 73 2D 77 69 66 69 61 6C 6C 69 61 6E 63 65  mas-wifialliance
2D 6F 72 67 3A 73 65 72 76 69 63 65 3A 57 46 41  -org:service:WFA
57 4C 41 4E 43 6F 6E 66 69 67 3A 31 0D 0A 4E 54  WLANConfig:1..NT
53 3A 20 73 73 64 70 3A 61 6C 69 76 65 0D 0A 0D  S: ssdp:alive...
0A                                               .

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
06/16-18:18:13.391269 192.168.1.1 -> 224.0.0.1
PROTO:002 TTL:1 TOS:0x0 ID:18704 IpLen:24 DgmLen:32
IP Options (1) => RTRALT
11 64 EE 9B 00 00 00 00                          .d......

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

WARNING: No preprocessors configured for policy 0.
06/16-18:18:16.300459 192.168.1.2 -> 224.0.0.252
PROTO:002 TTL:1 TOS:0x0 ID:9270 IpLen:24 DgmLen:32
IP Options (1) => RTRALT
16 00 09 03 E0 00 00 FC                          ........

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

RESULT

Thus Intrusion Detection System was demonstrated using Snort tool successfully.

Ex.No: 10 **INSTALLATION OF ROOTKITS AND STUDY ABOUT THE VARIETY OF OPTIONS**

AIM

To install rootkits and study about the variety of options.

PROCEDURE

Rootkit is a stealth type of malicious software designed to hide the existence of certain process from normal methods of detection and enables continued privileged access to a computer.

Step 1: Download Rootkit Tool from GMER website. www.gmer.net

Step 2: This displays the Processes, Modules, Services, Files, Registry, RootKit/Malwares, Autostart, CMD of local host.

Step 3: Select Processes menu and kill any unwanted process if any.

Step 4: Modules menu displays the various system files like .sys, .dll

Step 5: Services menu displays the complete services running with Autostart, Enable, Disable, System, Boot.

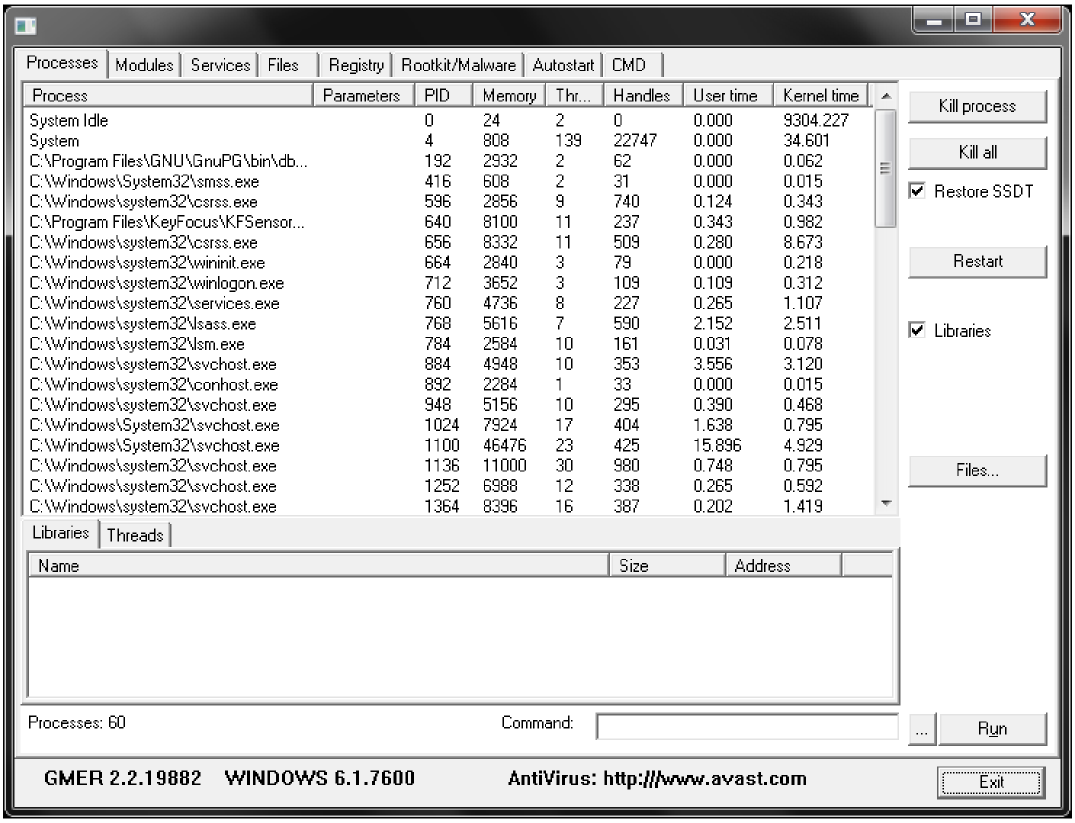
Step 6: Files menu displays full files on Hard-Disk volumes.

Step 7: Registry displays Hkey_Current_user and Hkey_Local_Machine.

Step 8: Rootkits/Malawares scans the local drives selected.

Step 9: Autostart displays the registry base Autostart applications.

Step 10: CMD allows the user to interact with command line utilities or Registry.



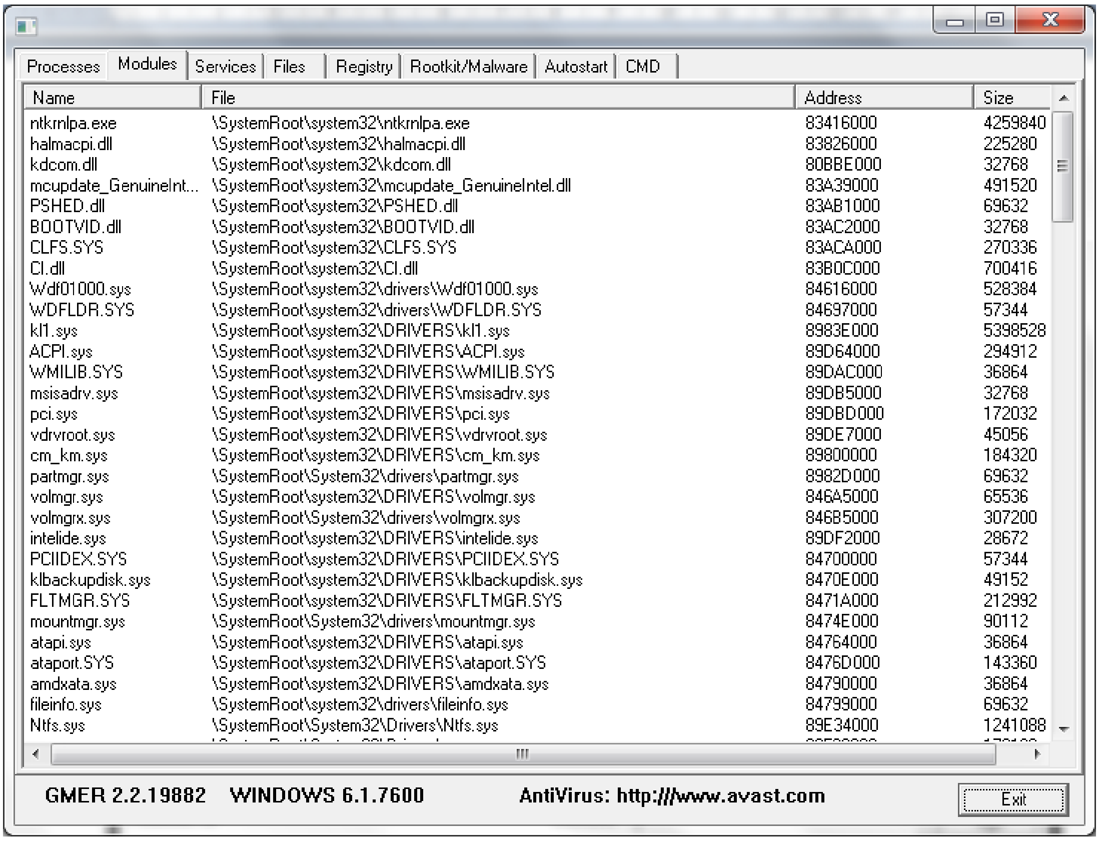
Process	Parameters	PID	Memory	Thr...	Handles	User time	Kernel time
System Idle		0	24	2	0	0.000	9304.227
System		4	808	139	22747	0.000	34.601
C:\Program Files\GNU\GnuPG\bin\db...		192	2932	2	62	0.000	0.062
C:\Windows\System32\smss.exe		416	608	2	31	0.000	0.015
C:\Windows\system32\csrss.exe		596	2856	9	740	0.124	0.343
C:\Program Files\KeyFocus\KFSensor...		640	8100	11	237	0.343	0.982
C:\Windows\system32\csrss.exe		656	8332	11	509	0.280	8.673
C:\Windows\system32\wininit.exe		664	2840	3	79	0.000	0.218
C:\Windows\system32\winlogon.exe		712	3652	3	109	0.109	0.312
C:\Windows\system32\services.exe		760	4736	8	227	0.265	1.107
C:\Windows\system32\lsass.exe		768	5616	7	590	2.152	2.511
C:\Windows\system32\lsm.exe		784	2584	10	161	0.031	0.078
C:\Windows\system32\svchost.exe		884	4948	10	353	3.556	3.120
C:\Windows\system32\conhost.exe		892	2284	1	33	0.000	0.015
C:\Windows\system32\svchost.exe		948	5156	10	295	0.390	0.468
C:\Windows\System32\svchost.exe		1024	7924	17	404	1.638	0.795
C:\Windows\System32\svchost.exe		1100	46476	23	425	15.896	4.929
C:\Windows\system32\svchost.exe		1136	11000	30	980	0.748	0.795
C:\Windows\system32\svchost.exe		1252	6988	12	338	0.265	0.592
C:\Windows\system32\svchost.exe		1364	8396	16	387	0.202	1.419

☑ Restore SSDT

☑ Libraries

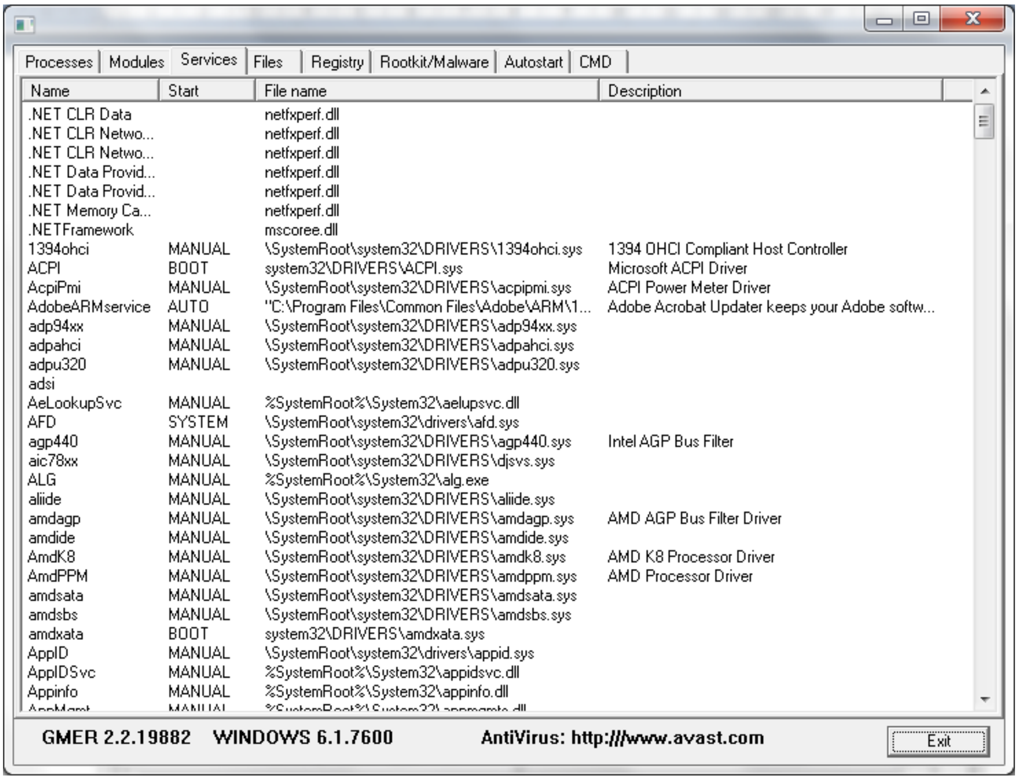
Processes: 60

GMER 2.2.19882 WINDOWS 6.1.7600 AntiVirus: http://www.avast.com



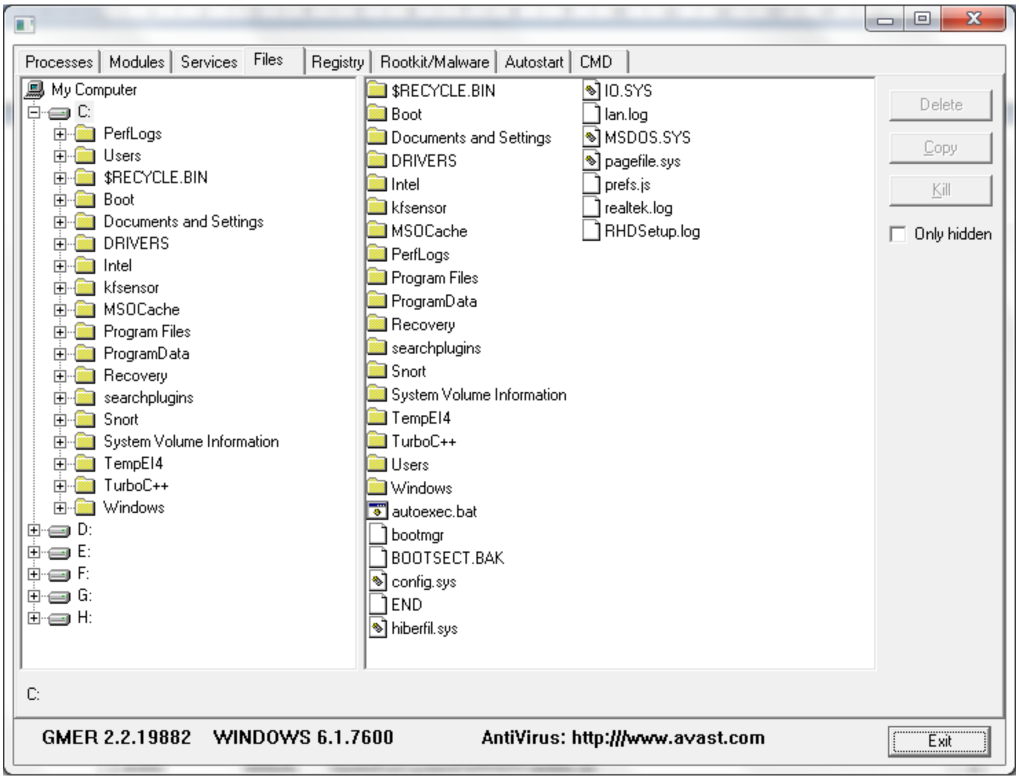
Name	File	Address	Size
ntkrnlpa.exe	\SystemRoot\system32\ntkrnlpa.exe	83416000	4259840
halmacpi.dll	\SystemRoot\system32\halmacpi.dll	83826000	225280
kdcom.dll	\SystemRoot\system32\kdcom.dll	80BBE000	32768
mcupdate_GenuineInt...	\SystemRoot\system32\mcupdate_GenuineIntel.dll	83A39000	491520
PSHED.dll	\SystemRoot\system32\PSHED.dll	83AB1000	69632
BOOTVID.dll	\SystemRoot\system32\BOOTVID.dll	83AC2000	32768
CLFS.SYS	\SystemRoot\system32\CLFS.SYS	83ACA000	270336
CI.dll	\SystemRoot\system32\CI.dll	83B0C000	700416
Wdf01000.sys	\SystemRoot\system32\drivers\Wdf01000.sys	84616000	528384
WDFLDR.SYS	\SystemRoot\system32\drivers\WDFLDR.SYS	84697000	57344
kl1.sys	\SystemRoot\system32\DRIVERS\kl1.sys	8983E000	5398528
ACPI.sys	\SystemRoot\system32\DRIVERS\ACPI.sys	89D64000	294912
WMILIB.SYS	\SystemRoot\system32\DRIVERS\WMILIB.SYS	89DAC000	36864
msisadrv.sys	\SystemRoot\system32\DRIVERS\msisadrv.sys	89DB5000	32768
pci.sys	\SystemRoot\system32\DRIVERS\pci.sys	89DBD000	172032
vdrvroot.sys	\SystemRoot\system32\DRIVERS\vdrvroot.sys	89DE7000	45056
cm_km.sys	\SystemRoot\system32\DRIVERS\cm_km.sys	89800000	184320
partmgr.sys	\SystemRoot\System32\drivers\partmgr.sys	8982D000	69632
volmgr.sys	\SystemRoot\system32\DRIVERS\volmgr.sys	846A5000	65536
volmgrx.sys	\SystemRoot\System32\drivers\volmgrx.sys	846B5000	307200
intelide.sys	\SystemRoot\system32\DRIVERS\intelide.sys	89DF2000	28672
PCIIDEX.SYS	\SystemRoot\system32\DRIVERS\PCIIDEX.SYS	84700000	57344
klbackupdisk.sys	\SystemRoot\system32\DRIVERS\klbackupdisk.sys	8470E000	49152
FLTMGR.SYS	\SystemRoot\system32\DRIVERS\FLTMGR.SYS	8471A000	212992
mountmgr.sys	\SystemRoot\System32\drivers\mountmgr.sys	8474E000	90112
atapi.sys	\SystemRoot\system32\DRIVERS\atapi.sys	84764000	36864
ataport.SYS	\SystemRoot\system32\DRIVERS\ataport.SYS	8476D000	143360
amdxata.sys	\SystemRoot\system32\DRIVERS\amdxata.sys	84790000	36864
fileinfo.sys	\SystemRoot\system32\drivers\fileinfo.sys	84799000	69632
Ntfs.sys	\SystemRoot\System32\Drivers\Ntfs.sys	89E34000	1241088

GMER 2.2.19882 WINDOWS 6.1.7600 AntiVirus: http://www.avast.com


Processes | Modules | Services | Files | Registry | Rootkit/Malware | Autostart | CMD

Name	Start	File name	Description
.NET CLR Data		netfxperf.dll	
.NET CLR Netwo...		netfxperf.dll	
.NET CLR Netwo...		netfxperf.dll	
.NET Data Provid...		netfxperf.dll	
.NET Data Provid...		netfxperf.dll	
.NET Memory Ca...		netfxperf.dll	
.NETFramework		mscoree.dll	
1394ohci	MANUAL	\SystemRoot\system32\DRIVERS\1394ohci.sys	1394 OHCI Compliant Host Controller
ACPI	BOOT	system32\DRIVERS\ACPI.sys	Microsoft ACPI Driver
AcpiPmi	MANUAL	\SystemRoot\system32\DRIVERS\acpipmi.sys	ACPI Power Meter Driver
AdobeARMservice	AUTO	"C:\Program Files\Common Files\Adobe\ARM\1...	Adobe Acrobat Updater keeps your Adobe softw...
adp94xx	MANUAL	\SystemRoot\system32\DRIVERS\adp94xx.sys	
adpahci	MANUAL	\SystemRoot\system32\DRIVERS\adpahci.sys	
adpu320	MANUAL	\SystemRoot\system32\DRIVERS\adpu320.sys	
adsi			
AeLookupSvc	MANUAL	%SystemRoot%\System32\aelupsvc.dll	
AFD	SYSTEM	\SystemRoot\system32\drivers\afd.sys	
agp440	MANUAL	\SystemRoot\system32\DRIVERS\agp440.sys	Intel AGP Bus Filter
aic78xx	MANUAL	\SystemRoot\system32\DRIVERS\djsvs.sys	
ALG	MANUAL	%SystemRoot%\System32\alg.exe	
aliide	MANUAL	\SystemRoot\system32\DRIVERS\aliide.sys	
amdagp	MANUAL	\SystemRoot\system32\DRIVERS\amdagp.sys	AMD AGP Bus Filter Driver
amdide	MANUAL	\SystemRoot\system32\DRIVERS\amdide.sys	
AmdK8	MANUAL	\SystemRoot\system32\DRIVERS\amdk8.sys	AMD K8 Processor Driver
AmdPPM	MANUAL	\SystemRoot\system32\DRIVERS\amdppm.sys	AMD Processor Driver
amdsata	MANUAL	\SystemRoot\system32\DRIVERS\amdsata.sys	
amdsbs	MANUAL	\SystemRoot\system32\DRIVERS\amdsbs.sys	
amdxata	BOOT	system32\DRIVERS\amdxata.sys	
AppID	MANUAL	\SystemRoot\system32\drivers\appid.sys	
AppIDSvc	MANUAL	%SystemRoot%\System32\appidsvc.dll	
Appinfo	MANUAL	%SystemRoot%\System32\appinfo.dll	

GMER 2.2.19882 WINDOWS 6.1.7600 AntiVirus: http://www.avast.com

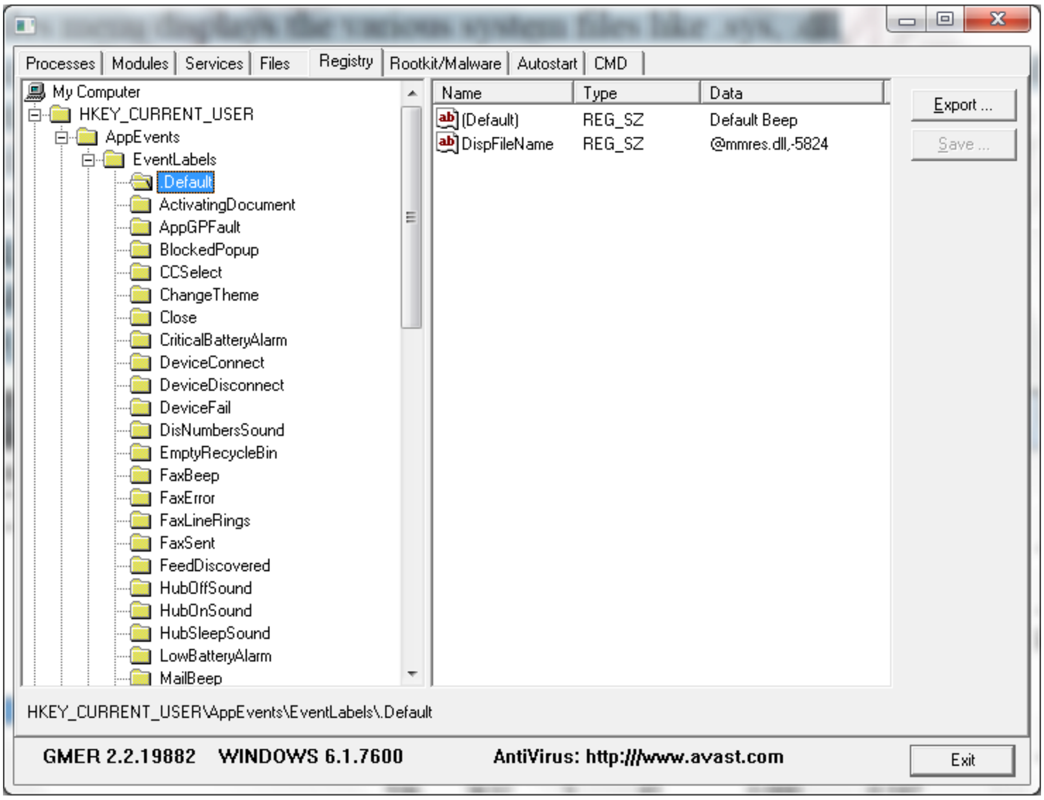

Processes | Modules | Services | Files | Registry | Rootkit/Malware | Autostart | CMD

My Computer
- C:
  - PerfLogs
  - Users
  - $RECYCLE.BIN
  - Boot
  - Documents and Settings
  - DRIVERS
  - Intel
  - kfsensor
  - MSOCache
  - Program Files
  - ProgramData
  - Recovery
  - searchplugins
  - Snort
  - System Volume Information
  - TempEI4
  - TurboC++
  - Windows
- D:
- E:
- F:
- G:
- H:

$RECYCLE.BIN
Boot
Documents and Settings
DRIVERS
Intel
kfsensor
MSOCache
PerfLogs
Program Files
ProgramData
Recovery
searchplugins
Snort
System Volume Information
TempEI4
TurboC++
Users
Windows
autoexec.bat
bootmgr
BOOTSECT.BAK
config.sys
END
hiberfil.sys
IO.SYS
lan.log
MSDOS.SYS
pagefile.sys
prefs.js
realtek.log
RHDSetup.log

C:

GMER 2.2.19882 WINDOWS 6.1.7600 AntiVirus: http://www.avast.com

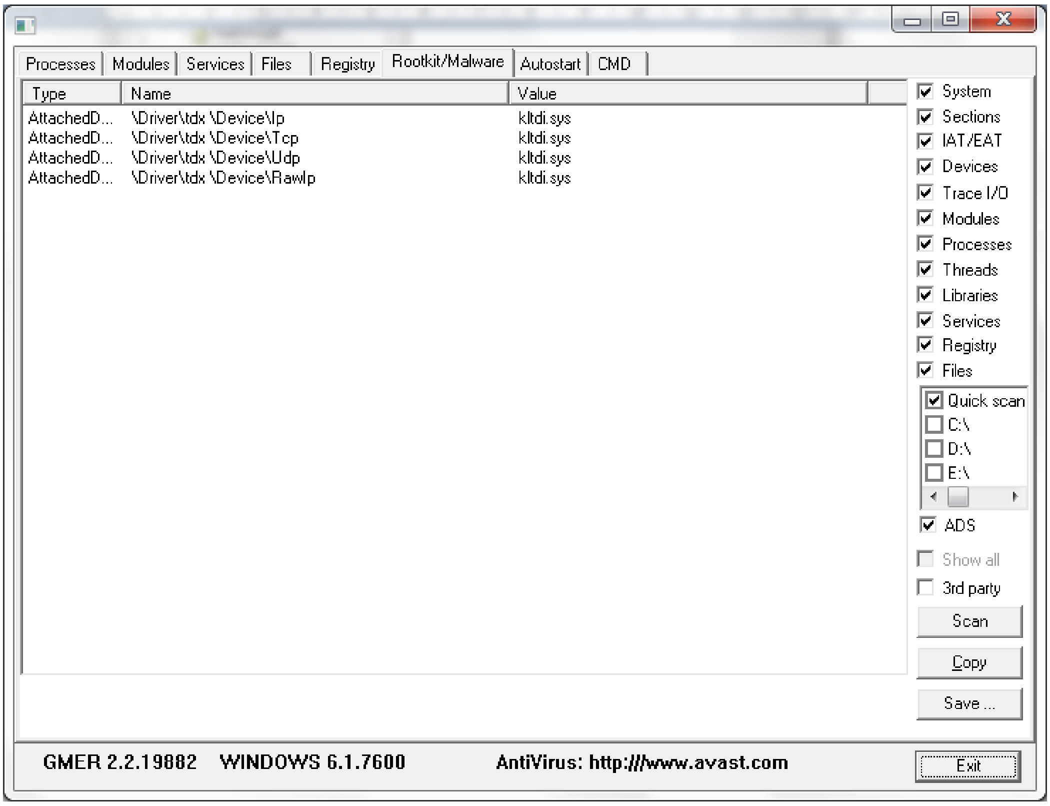


```
My Computer
  HKEY_CURRENT_USER
    AppEvents
      EventLabels
        .Default
        ActivatingDocument
        AppGPFault
        BlockedPopup
        CCSelect
        ChangeTheme
        Close
        CriticalBatteryAlarm
        DeviceConnect
        DeviceDisconnect
        DeviceFail
        DisNumbersSound
        EmptyRecycleBin
        FaxBeep
        FaxError
        FaxLineRings
        FaxSent
        FeedDiscovered
        HubOffSound
        HubOnSound
        HubSleepSound
        LowBatteryAlarm
        MailBeep
```

Name	Type	Data
(Default)	REG_SZ	Default Beep
DispFileName	REG_SZ	@mmres.dll,-5824

HKEY_CURRENT_USER\AppEvents\EventLabels\.Default

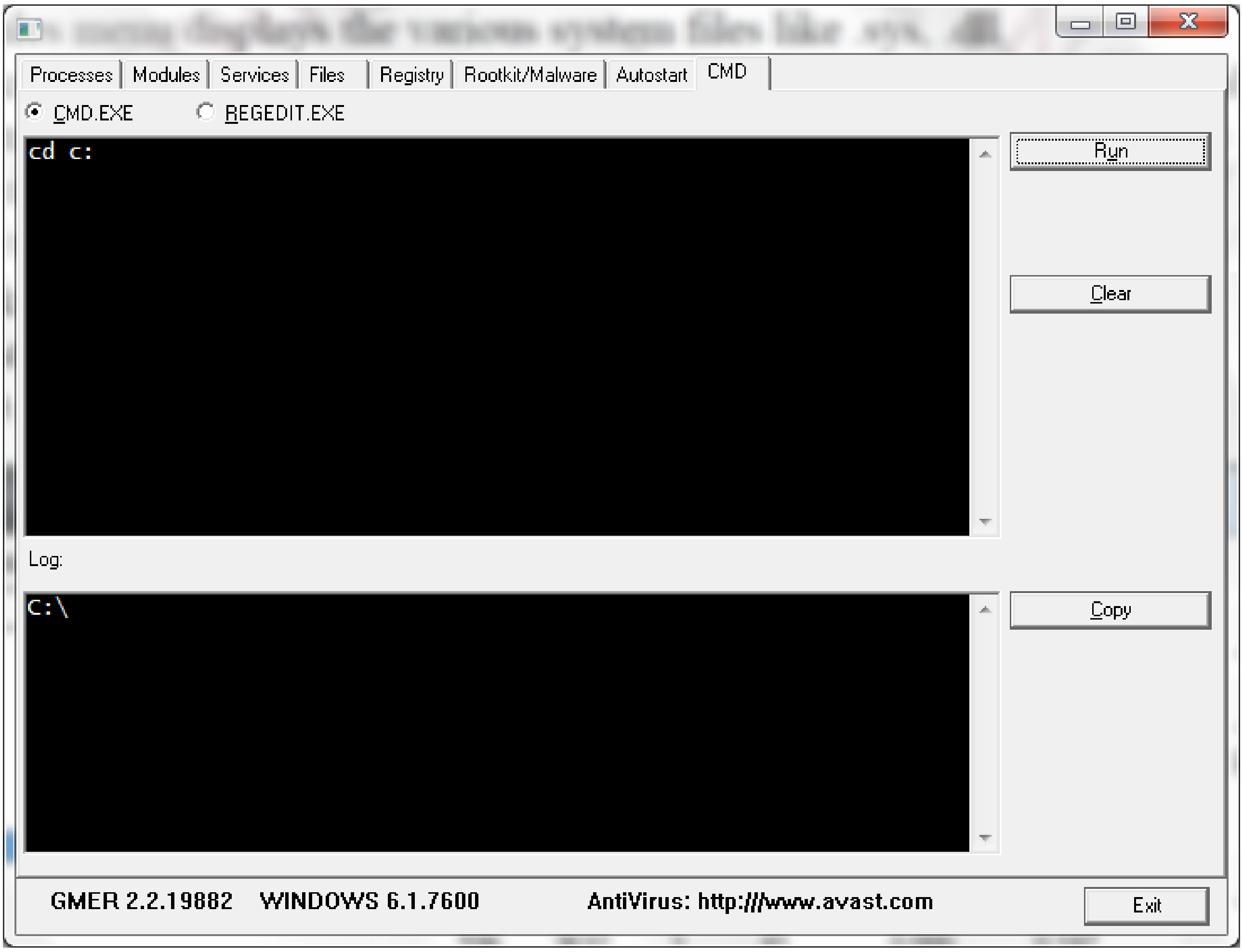
GMER 2.2.19882 WINDOWS 6.1.7600 AntiVirus: http://www.avast.com



Type	Name	Value
AttachedD...	\Driver\tdx \Device\Ip	kltdi.sys
AttachedD...	\Driver\tdx \Device\Tcp	kltdi.sys
AttachedD...	\Driver\tdx \Device\Udp	kltdi.sys
AttachedD...	\Driver\tdx \Device\RawIp	kltdi.sys

☑ System
☑ Sections
☑ IAT/EAT
☑ Devices
☑ Trace I/O
☑ Modules
☑ Processes
☑ Threads
☑ Libraries
☑ Services
☑ Registry
☑ Files
☑ Quick scan
□ C:\
□ D:\
□ E:\
☑ ADS
□ Show all
□ 3rd party

GMER 2.2.19882 WINDOWS 6.1.7600 AntiVirus: http://www.avast.com



● CMD.EXE ○ REGEDIT.EXE

```
cd c:
```

Log:

```
C:\
```

GMER 2.2.19882 WINDOWS 6.1.7600 AntiVirus: http:///www.avast.com

RESULT

Thus Rootkit was installed and various options were studied successfully.