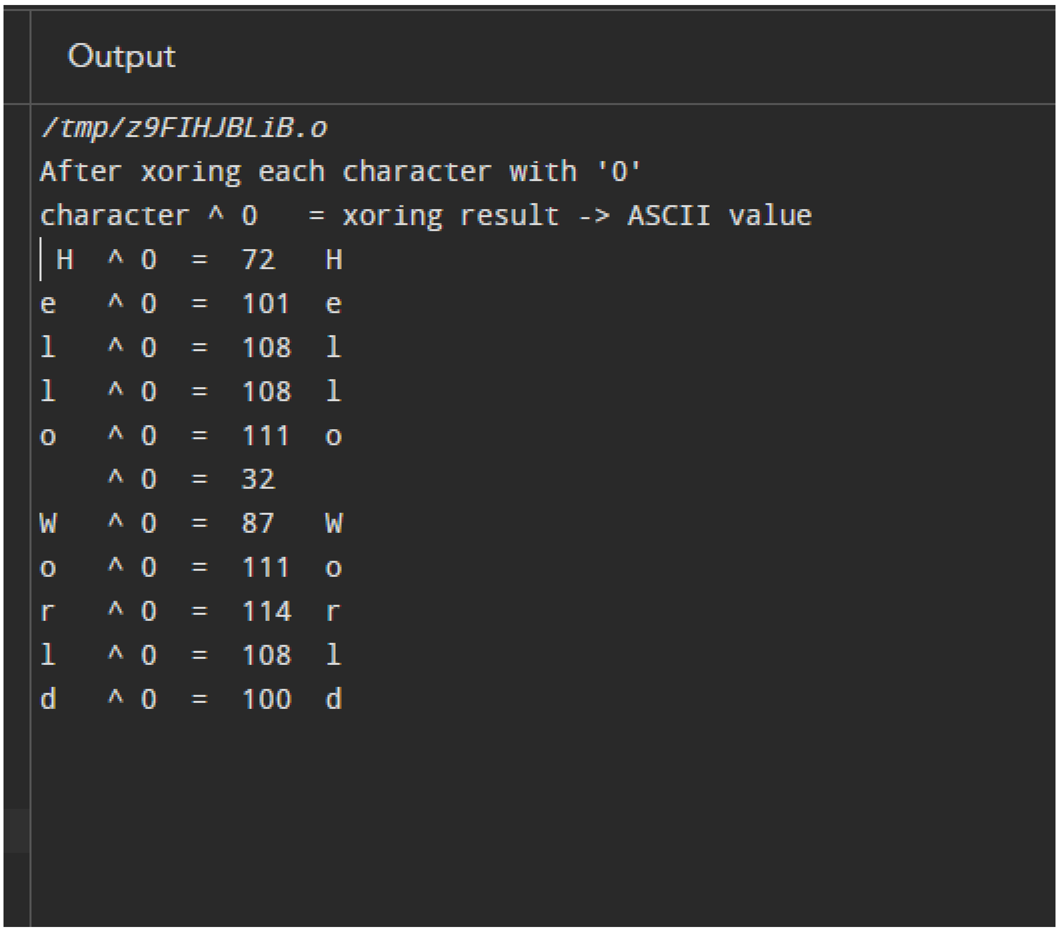
1. AIM: Write a C program that contains a string (char pointer) with a value \Hello World'.

The program should XOR each character in this string with 0 and displays the result.

Program:

```
#include<stdio.h>

#include<string.h>

#include<stdlib.h>

void main(){

        char *str = "Hello World";

        int l,i;

        //finding string length

        l = strlen(str);

        printf("After xoring each character with '0'\n");


        printf("character ^ 0 \t= xoring result -> ASCII value\n ");

        for(i=0; i<l; i++){

                printf("%c\t^ %d\t =  %d\t %c\n", str[i], 0, str[i]^0, (char)str[i]^0);

        }

}
```



Output

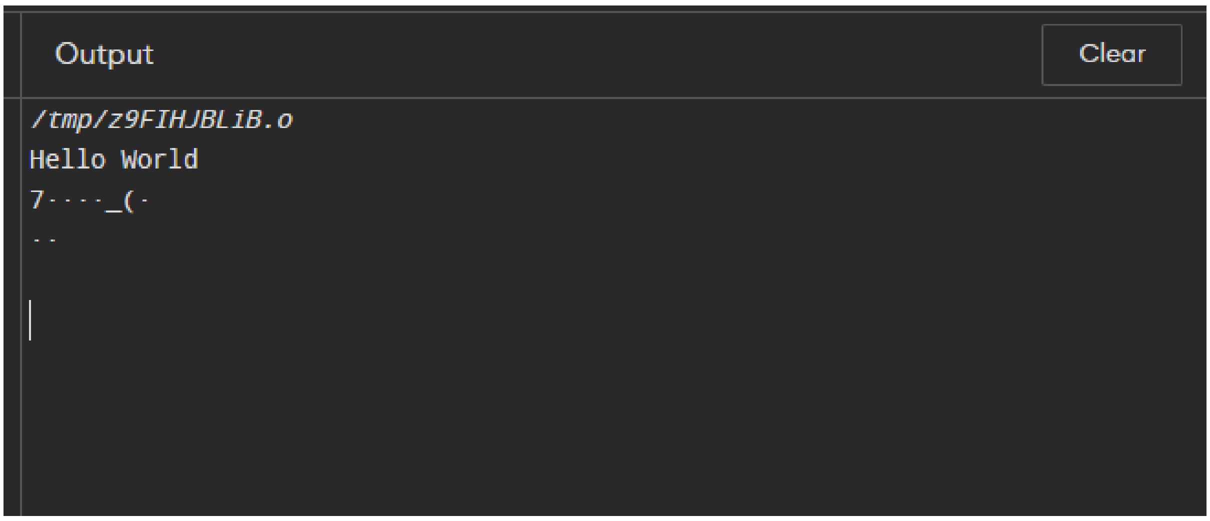
```
/tmp/z9FIHJBLiB.o
After xoring each character with '0'
character ^ 0   = xoring result -> ASCII value
 H  ^ 0  =  72   H
e   ^ 0  =  101  e
l   ^ 0  =  108  l
l   ^ 0  =  108  l
o   ^ 0  =  111  o
    ^ 0  =  32
W   ^ 0  =  87   W
o   ^ 0  =  111  o
r   ^ 0  =  114  r
l   ^ 0  =  108  l
d   ^ 0  =  100  d
```

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

Program:

```
#include<stdio.h>

#include<string.h>

#include<stdlib.h>


void main(){

   char str[]="Hello World";

   int i,len;
```

```
    len = strlen(str);
    for(i=0;i<len;i++){
        printf("%c",str[i]&127);
    }
    printf("\n");
    for(int i=0;i<len;i++){
        printf("%c",str[i]^127);
    }
    printf("\n");
    for(int i=0;i<len;i++){
        printf("%c",str[i]|127);
    }
    printf("\n");
}
```



```
/tmp/z9FIHJBLiB.o
Hello World
7····_(·
··
```

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

a) Ceaser Cipher

b) Substitution Cipher

c) Hill Cipher

Program:

a) Ceaser Cipher

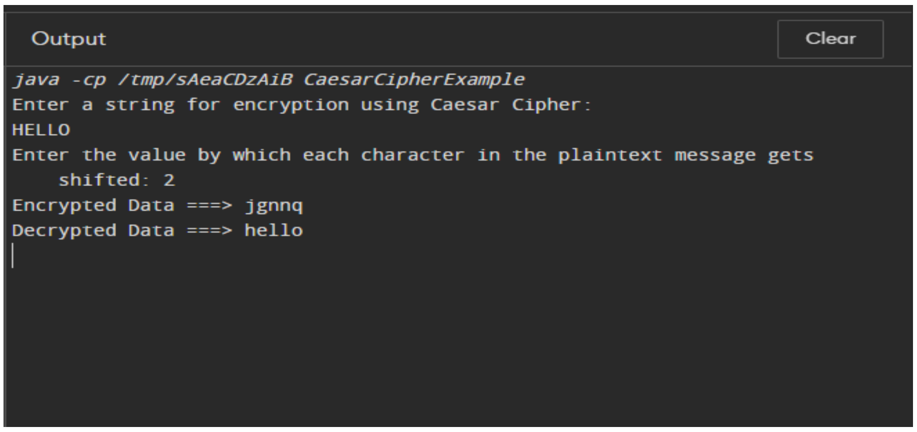
```
// import required classes and package, if any

import java.util.Scanner;


// create class CaesarCipherExample for encryption and decryption

public class CaesarCipherExample

{

   // ALPHABET string denotes alphabet from a-z

   public static final String ALPHABET = "abcdefghijklmnopqrstuvwxyz";


   // create encryptData() method for encrypting user input string with given shift key

   public static String encryptData(String inputStr, int shiftKey)

   {

      // convert inputStr into lower case

      inputStr = inputStr.toLowerCase();


      // encryptStr to store encrypted data

      String encryptStr = "";
```

```
    // use for loop for traversing each character of the input string
    for (int i = 0; i < inputStr.length(); i++)
    {
        // get position of each character of inputStr in ALPHABET
        int pos = ALPHABET.indexOf(inputStr.charAt(i));

        // get encrypted char for each char of inputStr
        int encryptPos = (shiftKey + pos) % 26;
        char encryptChar = ALPHABET.charAt(encryptPos);

        // add encrypted char to encrypted string
        encryptStr += encryptChar;
    }

    // return encrypted string
    return encryptStr;
}

// create decryptData() method for decrypting user input string with given shift key
public static String decryptData(String inputStr, int shiftKey)
{
    // convert inputStr into lower case
    inputStr = inputStr.toLowerCase();

    // decryptStr to store decrypted data
    String decryptStr = "";
    // use for loop for traversing each character of the input string
    for (int i = 0; i < inputStr.length(); i++)
    {

        // get position of each character of inputStr in ALPHABET
        int pos = ALPHABET.indexOf(inputStr.charAt(i));

        // get decrypted char for each char of inputStr
        int decryptPos = (pos - shiftKey) % 26;

        // if decryptPos is negative
        if (decryptPos < 0){
            decryptPos = ALPHABET.length() + decryptPos;
        }
        char decryptChar = ALPHABET.charAt(decryptPos);

        // add decrypted char to decrypted string
        decryptStr += decryptChar;
    }
    // return decrypted string
    return decryptStr;
}

// main() method start
public static void main(String[] args)
{
    // create an instance of Scanner class
```

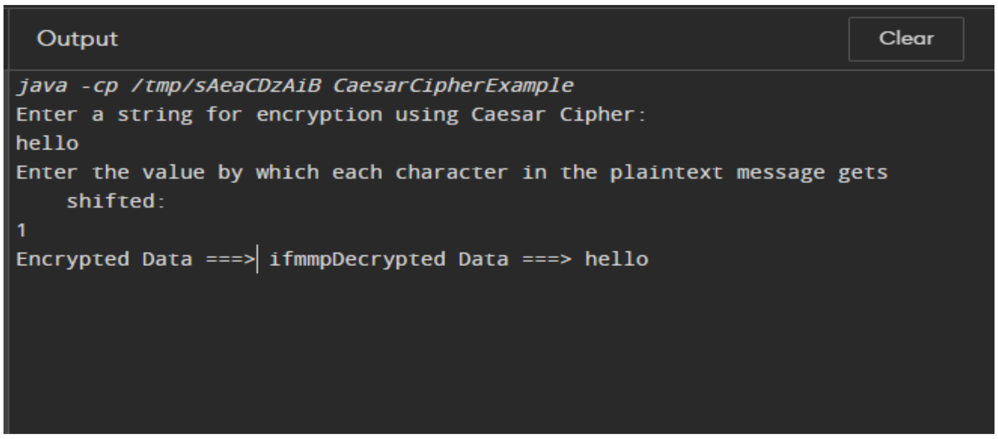
```
        Scanner sc = new Scanner(System.in);


        // take input from the user

        System.out.println("Enter a string for encryption using Caesar Cipher: ");

        String inputStr = sc.nextLine();


        System.out.println("Enter the value by which each character in the plaintext message
gets shifted: ");

        int shiftKey = Integer.valueOf(sc.nextLine());


        System.out.println("Encrypted Data ===> "+encryptData(inputStr, shiftKey));

        System.out.println("Decrypted Data ===> "+decryptData(encryptData(inputStr, shiftKey),
shiftKey));


        // close Scanner class object

        sc.close();

    }

}
```

Output:

```
Enter any String: Hello World

Enter the Key: 5

Encrypted String is: mjqqtebtwqi

Decrypted String is: Hello World
```



```
java -cp /tmp/sAeaCDzAiB CaesarCipherExample
Enter a string for encryption using Caesar Cipher:
HELLO
Enter the value by which each character in the plaintext message gets
    shifted: 2
Encrypted Data ===> jgnnq
Decrypted Data ===> hello
```



```
java -cp /tmp/sAeaCDzAiB CaesarCipherExample
Enter a string for encryption using Caesar Cipher:
hello
Enter the value by which each character in the plaintext message gets
    shifted:
1
Encrypted Data ===> ifmmpDecrypted Data ===> hello
```

b) Substitution cipher

Program:

import java.io.*;

import java.util.*;

public class SubstitutionCipher{

static Scanner sc = new Scanner(System.in);

static BufferedReader br = new BufferedReader(new InputStreamReader(System.in)); public

static void main(String[] args) throws IOException {

// TODO code application logic here

String a = "abcdefghijklmnopqrstuvwxyz";

String b ="zyxwvutsrqponmlkjihgfedcba";

System.out.print("Enter any string: "); String str = br.readLine();

String decrypt = "";

char c;

for(int i=0;i<str.length();i++)

{

c = str.charAt(i); int

j = a.indexOf(c);

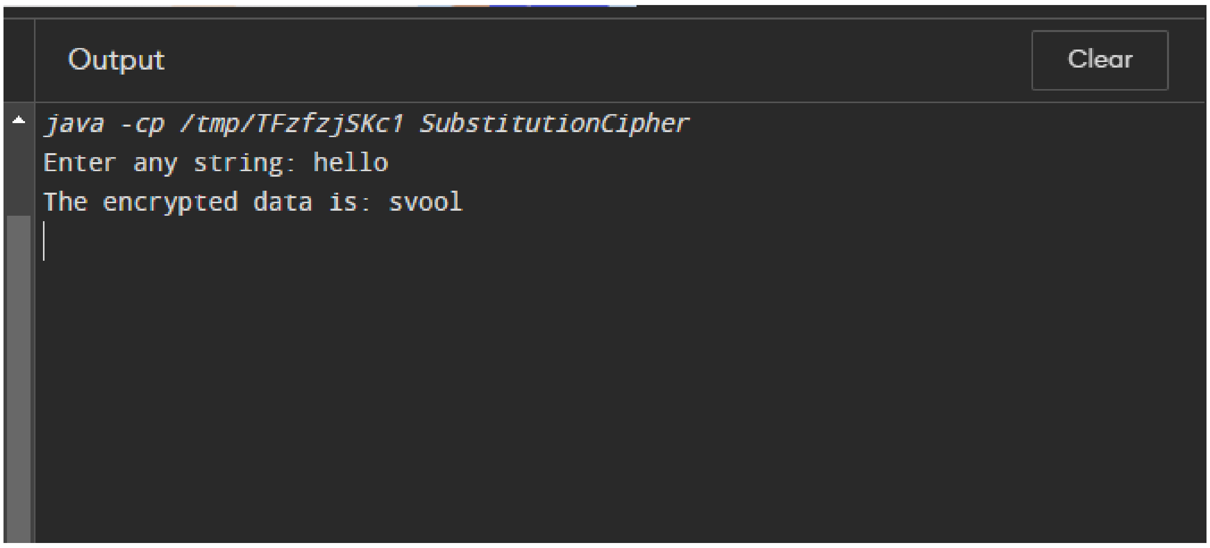
decrypt = decrypt+b.charAt(j);

}

System.out.println("The encrypted data is: " +decrypt);

}

}



```
java -cp /tmp/TFzfzjSKc1 SubstitutionCipher
Enter any string: hello
The encrypted data is: svool
```

c) Hill Cipher

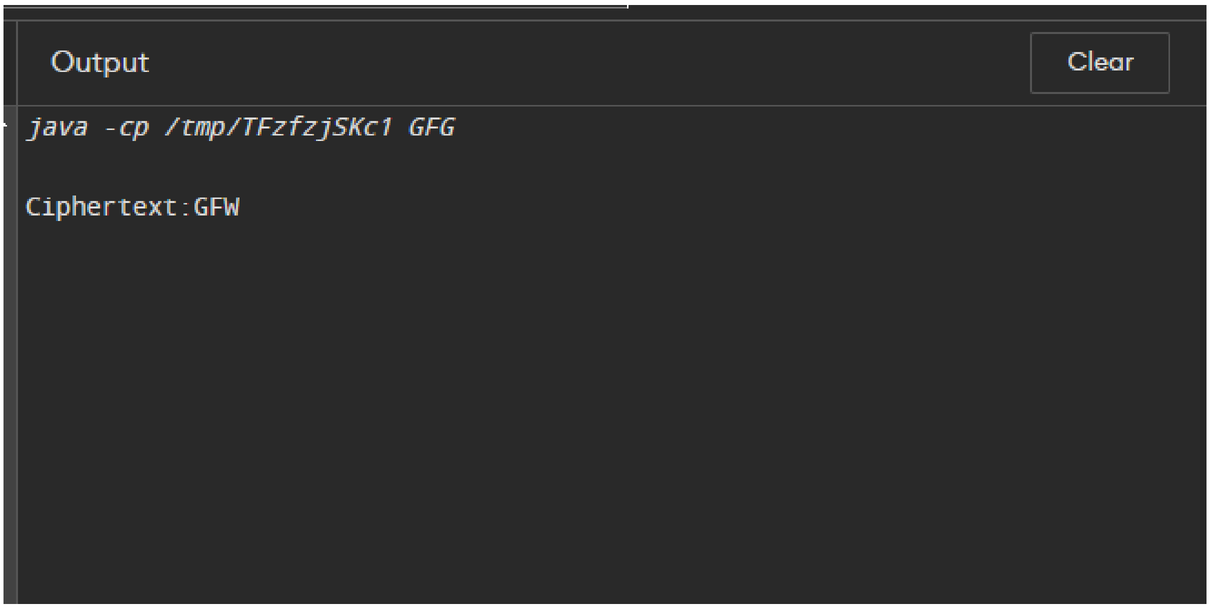
// Java code to implement Hill Cipher

class GFG

{

```
// Following function generates the
// key matrix for the key string
static void getKeyMatrix(String key, int keyMatrix[][])
{
        int k = 0;
        for (int i = 0; i < 3; i++)
        {
                for (int j = 0; j < 3; j++)
                {
                        keyMatrix[i][j] = (key.charAt(k)) % 65;
                        k++;
                }
        }
}

// Following function encrypts the message
static void encrypt(int cipherMatrix[][],
                        int keyMatrix[][],
                        int messageVector[][])
{
        int x, i, j;
        for (i = 0; i < 3; i++)
        {
                for (j = 0; j < 1; j++)
                {
                        cipherMatrix[i][j] = 0;
```

```
                    for (x = 0; x < 3; x++)
                    {
                        cipherMatrix[i][j] +=
                            keyMatrix[i][x] * messageVector[x][j];
                    }

                    cipherMatrix[i][j] = cipherMatrix[i][j] % 26;
                }
            }
        }

        // Function to implement Hill Cipher
        static void HillCipher(String message, String key)
        {
            // Get key matrix from the key string
            int [][]keyMatrix = new int[3][3];
            getKeyMatrix(key, keyMatrix);

            int [][]messageVector = new int[3][1];

            // Generate vector for the message
            for (int i = 0; i < 3; i++)
                messageVector[i][0] = (message.charAt(i)) % 65;

            int [][]cipherMatrix = new int[3][1];

            // Following function generates
            // the encrypted vector
            encrypt(cipherMatrix, keyMatrix, messageVector);

            String CipherText="";

            // Generate the encrypted text from
            // the encrypted vector
            for (int i = 0; i < 3; i++)
                CipherText += (char)(cipherMatrix[i][0] + 65);

            // Finally print the ciphertext
            System.out.print(" Ciphertext:" + CipherText);
        }

        // Driver code
        public static void main(String[] args)
        {
            // Get the message to be encrypted
            String message = "HEY";

            // Get the key
            String key = "GYBNQKURP";

            HillCipher(message, key);
            }
        }
```



```
java -cp /tmp/TFzfzjSKc1 GFG

Ciphertext:GFW
```

Output:

Ciphertext:GFW