CSS LAB

1)CAESAR CIPHER

PROGRAM:

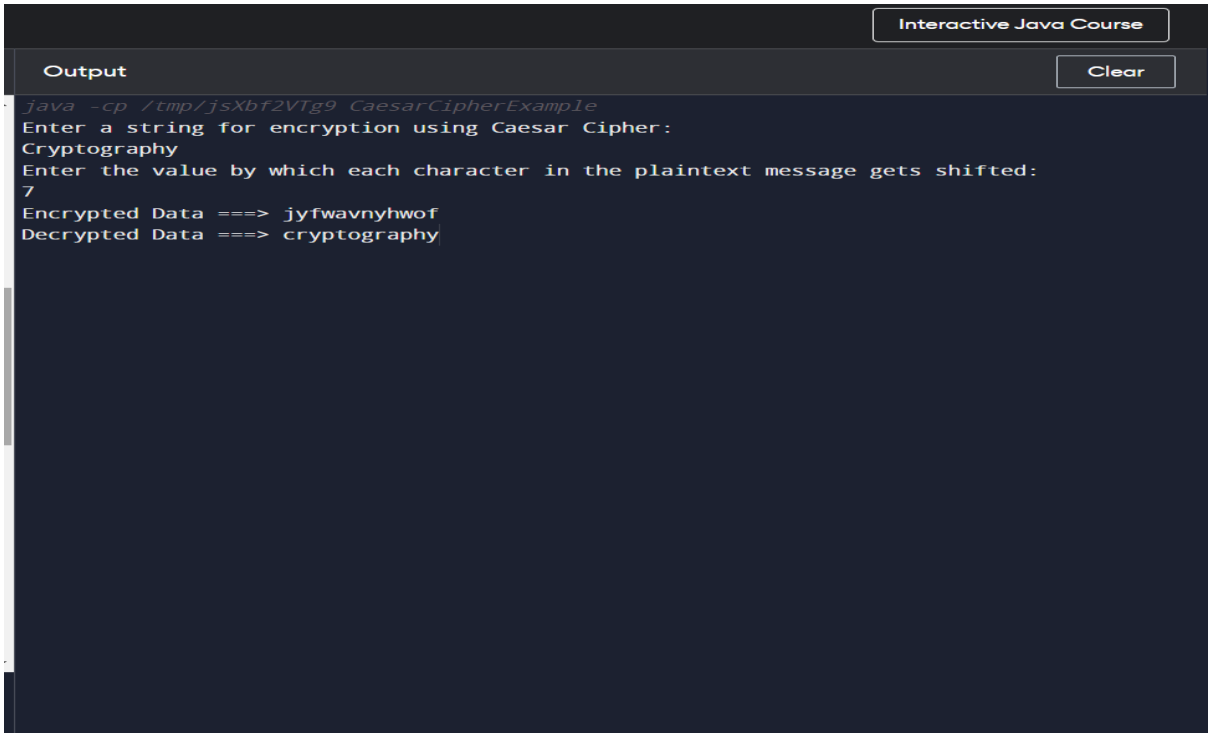
```
import java.util.Scanner;

// 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:



PROGRAM 2 :

COLUMNAR TRANSPOSITION

```
import java.util.*;


public class columnartransposition {

    public static void main(String[] args) {

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

```
    System.out.println("Enter a text: ");
    String txt = input.nextLine();
    System.out.println("Enter a num of columns: ");
    int num = input.nextInt();
    String cipher = cipherTrans(txt, num);
    System.out.println("Coded: " + cipher);
    String decipher = decipherTrans(cipher, num);
    System.out.println("Decoded: " + decipher);
    input.close();
 }

//ENCRYPTION
 private static String cipherTrans(String txt, int num) {
    String output = "";

    int num2 = (int) Math.ceil(txt.length()*1.0/num);

    char[][] buffer = new char[num2][num];
    int k = 0;

    for (int i = 0; i < num2; i++) {
        for (int j = 0; j < num; j++, k++) {
            if (txt.length() > k) {
                buffer[i][j] = txt.charAt(k);
            }
        }
    }

    for (int j = 0; j < num; j++) {
        for (int i = 0; i < num2; i++) {
```

```
                output += buffer[i][j];
            }
        }
        return output;
    }

    //DECRYPTION
    private static String decipherTrans(String txt, int num) {
        int num2 = (int) Math.ceil(txt.length() * 1.0 / num);

        char[][] buffer = new char[num2][num];
        int k = 0;
        for (int i = 0; i < num; i++) {
            for (int j = 0; j < num2; j++, k++) {
                if (txt.length() > k) {
                    buffer[j][i] = txt.charAt(k);
                }
            }
        }

        String output = "";
        for (int i = 0; i < num2; i++){
            for (int j = 0; j < num; j++){
                output += buffer[i][j];
            }
        }
        return output;
    }
}
```

Output

Clear

```
java -cp /tmp/jsXbf2VTg9 columnartransposition
Enter a text: attack is from the south
Enter a num of columns:
5
Coded: akftot rhutioetasm hc  s
Decoded: attack is from the south
```

29°C Haze ENG 20:41:08 30-01-2023