NAME: SUBHAPREET PATRO

ROLL NO.: 2211CS10547

GROUP: 3

Week-2

Aim:

Program for the execution of Caesar Cipher in Java.

Description:

The Caesar cipher is the simplest and oldest method of cryptography. The Caesar cipher method is based on a mono-alphabetic cipher and is also called a shift cipher or additive cipher. Julius Caesar used the shift cipher (additive cipher) technique to communicate with his officers. For this reason, the shift cipher technique is called the Caesar cipher. The Caesar cipher is a kind of replacement (substitution) cipher, where all letter of plain text is replaced by another letter.

Let's take an example to understand the Caesar cipher, suppose we are shifting with 1, then A will be replaced by B, B will be replaced by C, C will be replaced by D, D will be replaced by C, and this process continues until the entire plain text is finished.

Caesar ciphers is a weak method of cryptography. It can be easily hacked. It means the message encrypted by this method can be easily decrypted.

Plaintext: It is a simple message written by the user.

Ciphertext: It is an encrypted message after applying some technique.

The formula of encryption is:

$E_n(x) = (x + n) \bmod 26$

The formula of decryption is:

$D_n(x) = (xi - n) \bmod 26$

If any case (Dn) value becomes negative (-ve), in this case, we will add 26 in the negative value.

Where,

E denotes the encryption
D denotes the decryption
x denotes the letters value
n denotes the key value (shift value)

Example:
Text : ABCDEFGHIJKLMNOPQRSTUVWXYZ
Shift: 23
Cipher: XYZABCDEFGHIJKLMNOPQRSTUVW

Text : ATTACKATONCE
Shift: 4
Cipher: EXXEGOEXSRGI

Program:

```java
import java.io.BufferedReader;
import java.io.IOException;
import java.io.InputStreamReader;
import java.util.Scanner;


public class CeasarCipher {
    static Scanner sc = new Scanner(System.in);
    static BufferedReader br = new BufferedReader(new
InputStreamReader(System.in));
    public static void main(String[] args) throws IOException {
        System.out.println("Enter any string: ");
        String str=br.readLine();
        System.out.println("Enter the key: ");
        int key=sc.nextInt();
        String encrpted = encrpyt(str,key);
        System.out.println("\nEncrypted String is: "+encrpted);
        System.out.println();
        String decrpted = decrpyt(encrpted,key);
        System.out.println("Decrypted String is: "+decrpted);
        System.out.println("\n");
    }
    public static String encrpyt(String str,int key){
        String encrypted="";
        for(int i=0;i<str.length();i++){
            int c=str.charAt(i);
```

```java
            if (Character.isUpperCase(c)) {
                c = c + (key % 26);
                if (c > 'Z')
                    c = c - 26;
            } else if (Character.isLowerCase(c)) {
                c = c + (key % 26);
                if (c > 'z')
                    c = c - 26;
            }
            encrypted+=(char) c;
        }
        return encrypted;
    }
    public static String decrpyt(String str,int key){
        String decrypted="";
        for(int i=0;i<str.length();i++){
            int c=str.charAt(i);
            if (Character.isUpperCase(c)) {
                c = c - (key % 26);
                if (c < 'A')
                    c = c + 26;
            } else if (Character.isLowerCase(c)) {
                c = c - (key % 26);
                if (c < 'a')
                    c = c + 26;
            }
            decrypted+=(char) c;
```

```
        }

        return decrypted;

    }

}
```

Output: