Name: B.INDU

Roll No: 2211CS010665

Group: Group-6

WEEK-2

Title: java program to implement Ceaser-Cipher encryption technique

Ceaser-Cipher Encryption:

The Caesar cipher is a simple encryption technique that was used by Julius Caesar to send secret messages to his allies. It works by shifting the letters in the plaintext message by a certain number of positions, known as the "shift" or "key". The Caesar Cipher technique is one of the earliest and simplest methods of encryption techniques.

It's simply a type of substitution cipher, i.e., each letter of a given text is replaced by a letter with a fixed number of positions down the alphabet. For example with a shift of 1, A would be replaced by B, B would become C, and so on. The method is apparently named after Julius Caesar, who apparently used it to communicate with his officials**.**

- For each plaintext letter p, substitute the cipher text letter c such that

$C = E(p) = (p+3) \bmod 26$

- A shift may be any amount, so that general Caesar algorithm is

$C = E(p) = (p+k) \bmod 26$

- Where k takes on a value in the range 1 to 25. The decryption algorithm is simply

$P = D(C) = (C-k) \bmod 26$

Where:

- P is the plaintext letter (0–25 for A–Z)
- C is the ciphertext letter (0–25 for A–Z).
- k is the shift key.

Example:

Key (Shift Value):

k=3 Plaintext: HELLO Encryption:

H→ K

E → H

L → O

L → O

O → R

Ciphertext: KHOOR

Decryption: Reverse the process with k=−3**.**

Code:

```
 import java.io.BufferedReader;
 import java.io.IOException;
 import java.io.InputStreamReader;
 import java.util.Scanner;
public class CeaserCipher
{
       static Scanner sc=new Scanner(System.in);
       static BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
  public static void main(String[] args) throws IOException
    {
// TO DO code application logic here
          System.out.print("Enter any String: ");
          String str = br.readLine();
          System.out.print("\nEnter the Key: "); int key= sc.nextInt();
          String encrypted = encrypt(str, key);
          System.out.println("\nEncrypted String is: " +encrypted);
          System.out.println();
          String decrypted = decrypt(encrypted, key);
          System.out.println("\nDecrypted String is:"+decrypted);
          System.out.println("\n");
    }
           public static String encrypt(String str, int key)
               {
```

```
        String encrypted ="";
           for(int i = 0; i < str.length(); i++)
            {
                int c= str.charAt(i);
                  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 decrypt(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: