# Practical 1

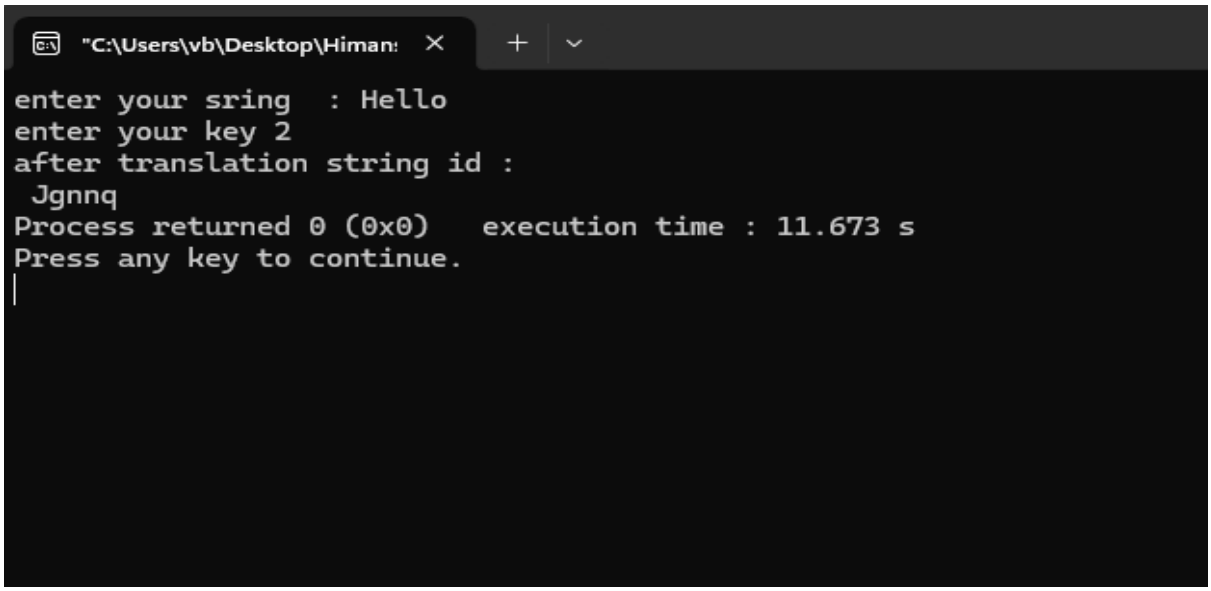
**AIM : Implement Ceasar cipher. It is a substitution cipher. Analyze the strength of the cipher in terms of brute force attack and cryptanalysis attack. Test Case: Hello, Welcome. The key used is 3.**

**Code :**

```
import java.util.Scanner;
public class CaesarCipher {
   public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);



      System.out.print("Enter your string: ");
      String input = scanner.next();



      System.out.print("Enter your key: ");
      int k = scanner.nextInt();

      char[] transtext = new char[input.length()];
      for (int i = 0; i < input.length(); i++) {
         char ch = input.charAt(i);

         if (Character.isUpperCase(ch)) {
            transtext[i] = (char) (((ch - 'A' + k) % 26 + 26) % 26 + 'A');
         }
         else if (Character.isLowerCase(ch)) {
            transtext[i] = (char) (((ch - 'a' + k) % 26 + 26) % 26 + 'a');
         }
         else {
```

```
            transtext[i] = ch;
          }
        }

        // 4. Output the result
        System.out.println("After translation string is:");
        System.out.println(new String(transtext));

        scanner.close();
    }
}
```

**Output :**



# Practical 2

**AIM : Implement Brute Force approach to Break Caesar Cipher Encryption.**

**CODE :**

```
class practical2 {
   public static void main(String[] args) {
      String cipherText ="KHOOR ZRUOG";
      for(int i=1 ;i<=25;i++)
      {
         String decrypted = Decrypt(cipherText  ,i);
         System.out.println( "key" + i + ":" + decrypted);
      }
   }
   public static String  Decrypt(String txt , int shift  )
   {
      StringBuilder result = new StringBuilder();
      for( char ch : txt.toCharArray())
      {
         if(Character.isUpperCase(ch))
         {
            char decryptedChar = (char)((ch -'A' -shift +26)%26 +'A');
            result.append(decryptedChar);
         }

        else  if(Character.isLowerCase(ch))
         {
            char decryptedChar = (char)((ch -'a' -shift +26)%26 +'a');
            result.append(decryptedChar);
```

```
        }
        else
        result.append(ch);
      }
      return result.toString();
   }
}
```

**Output :**

```
PS C:\Users\vb\Desktop\Himanshu IT--3 CNS>  & 'C:\Program Files\Java\jdk-20\bin\java.exe' '-XX:+ShowCodeDeta
sInExceptionMessages' '-cp' 'C:\Users\vb\AppData\Roaming\Code\User\workspaceStorage\05d05e17638e4c2b50041021
2dbd0b\redhat.java\jdt_ws\Himanshu IT--3 CNS_27ced9ee\bin' 'practical2'
key1:JGNNQ YQTNF
key2:IFMMP XPSME
key3:HELLO WORLD
key4:GDKKN VNQKC
key5:FCJJM UMPJB
key6:EBIIL TLOIA
key7:DAHHK SKNHZ
key8:CZGGJ RJMGY
key9:BYFFI QILFX
key10:AXEEH PHKEW
key11:ZWDDG OGJDV
key12:YVCCF NFICU
key13:XUBBE MEHBT
key14:WTAAD LDGAS
key15:VSZZC KCFZR
key16:URYYB JBEYQ
key17:TQXXA IADXP
key18:SPWWZ HZCWO
key19:ROVVY GYBVN
key20:QNUUX FXAUM
key21:PMTTW EWZTL
key22:OLSSV DVYSK
key23:NKRRU CUXRJ
key24:MJQQT BTWQI
key25:LIPPS ASVPH
PS C:\Users\vb\Desktop\Himanshu IT--3 CNS>
```

# Practical 3

**AIM : Implement rail Fence cipher. It is permutation cipher. Analyze the strength of the cipher in terms of cryptanalysis. Test case : MEETME Cipher text : MEMETE**

**CODE :**

```
public class practical3
  {
     public static void main(String[] args) {
    String message = "meet me after toga party";
    StringBuilder even = new StringBuilder();
    StringBuilder odd = new StringBuilder();
    char[] result = message.toCharArray();
    for (int i = 0; i < result.length; i++)
        {
         if (i % 2 == 0)   even.append(result[i]);
         else  odd.append(result[i]);
        }
   String ans = even.toString() + odd.toString();
   System.out.println("answer is :" + ans);
  }
}
```

**Output :**

# Practical 4

**AIM : Implement Hill cipher. It is a substitution cipher. Analyze the strength of the cipher in terms of brute force attack and cryptanalysis attack. Suggest one way to improve and strengthen the cipher and analyze with respect to cryptanalysis attack. Plaintext = Hello Himansh**

**Ciphertext = TAS**

**Key K = 6 24 1**

**13 16 10**

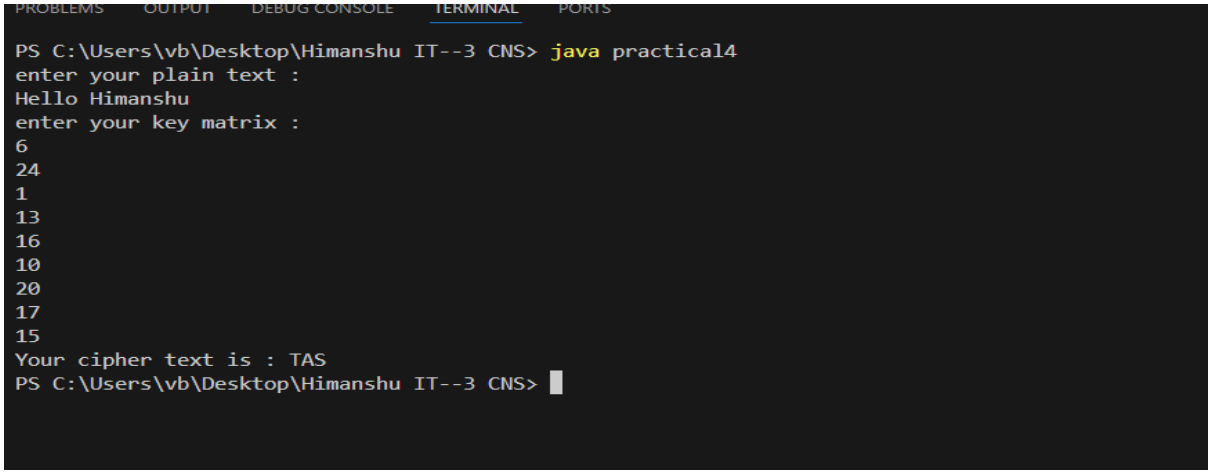
**20 17 15**

**CODE :**

```
import java.util.*;
public class practical4 {
  public static void main(String[] args)
  {
    Scanner sc = new Scanner(System.in);
    System.out.println("enter your plain text :");
    String plaintext = sc.nextLine();
    System.out.println("enter your key matrix :");
    int[][] matrix = new int[3][3];

    for (int i = 0; i < 3; i++) {
      for (int j = 0; j < 3; j++) {
        matrix[i][j] = sc.nextInt();
      }
    }

    plaintext = plaintext.toUpperCase().replaceAll("[^A-Z]", "");
```

```
    while (plaintext.length() % 3 != 0) {
      plaintext += "X";
    }

    int block[] = new int[3];
    StringBuilder cipherText = new StringBuilder();
    for (int i = 0; i < plaintext.length(); i += 3) {
      for (int j = 0; j < 3; j++) {
        block[j] = plaintext.charAt(i + j) - 'A';
      }
    }
    int result[] = new int[3];

    for (int i = 0; i < 3; i++) {
      for (int j = 0; j < 3; j++) {
        result[i] += matrix[i][j] * block[j];
      }
      result[i] %= 26;
    }
    for (int i = 0; i < 3; i++) {
      cipherText.append((char) (result[i] + 'A'));
    }
    System.out.println("Your cipher text is : " + cipherText);
    sc.close();
  }
}
```

**Output :**


PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\vb\Desktop\Himanshu IT--3 CNS> java practical4
enter your plain text :
Hello Himanshu
enter your key matrix :
6
24
1
13
16
10
20
17
15
Your cipher text is : TAS
PS C:\Users\vb\Desktop\Himanshu IT--3 CNS>