## Id no:vu3t4s2021006 EXPERIMENT NO-3 Security Lab

**Aim: Breaking the Mono-alphabetic Substitution Cipher using Frequency analysis method.**

**Theory:**

Monoalphabetic cipher is **a substitution cipher in which for a given key, the cipher alphabet for each plain alphabet is fixed throughout the encryption process**. For example, if 'A' is encrypted as 'D', for any number of occurrence in that plaintext, 'A' will always get encrypted to

'D'.

One of the way to break a monoalphabetic substitution cipher is to perform a frequency analysis of the characters in the ciphertext.

**ALGORITHM:**

1. Compare the frequencies obtained with the frequencies of each letter of the alphabet in english plaintext.

2.Match the character with the highest frequency in the ciphertext with the character with the highest frequency in the english language

3.Do the same thing with the next most frequent, then with the next until you have a complete mapping between the frequencies of the characters in the ciphertext and the frequencies of the letters in english plaintext.

**Program:**

import java.util.\*;

class GFG{

static void printString(String S, int N)

{

String []plaintext = new String[5];

int freq[] = new int[26];

int freqSorted[] = new int[26];

int Used[] = new int[26];

for (int i = 0; i < N; i++) {

if (S.charAt(i) != ' ') {

freq[S.charAt(i) - 'A']++;

**TE-IT-B VPPCoE Page No: 01**

}

}

for (int i = 0; i < 26; i++) {

freqSorted[i] = freq[i];

}

String T = "ETAOINSHRDLCUMWFGYPBVKJXQZ";

Arrays.sort(freqSorted);

freqSorted= reverse(freqSorted);

for (int i = 0; i < 5; i++) {

int ch = -1;

for (int j = 0; j < 26; j++) {

if (freqSorted[i] == freq[j] && Used[j] == 0) {

Used[j] = 1;

ch = j;

break;

}

}

if (ch == -1)

break;

int x = T.charAt(i) - 'A';

x = x - ch;

String curr = "";

for (int k = 0; k < N; k++) {

if (S.charAt(k) == ' ') {

curr += (char)' ';

continue;

}

int y = S.charAt(k) - 'A';

y += x;

if (y < 0)

y += 26;

if (y > 25)

y -= 26;

curr += (char)('A' + y);

}

plaintext[i] = curr;

}

for (int i = 0; i < 5; i++) {

System.out.print(plaintext[i] +"\n");}}

**TE-IT-B VPPCoE Page No:02**

static int[] reverse(int a[]) {

int i, n = a.length, t;

for (i = 0; i < n / 2; i++) {

t = a[i];

a[i] = a[n - i - 1];

a[n - i - 1] = t;

}

return a;

}

public static void main(String[] args)

{

String S = "B TJNQMF NFTTBHF";

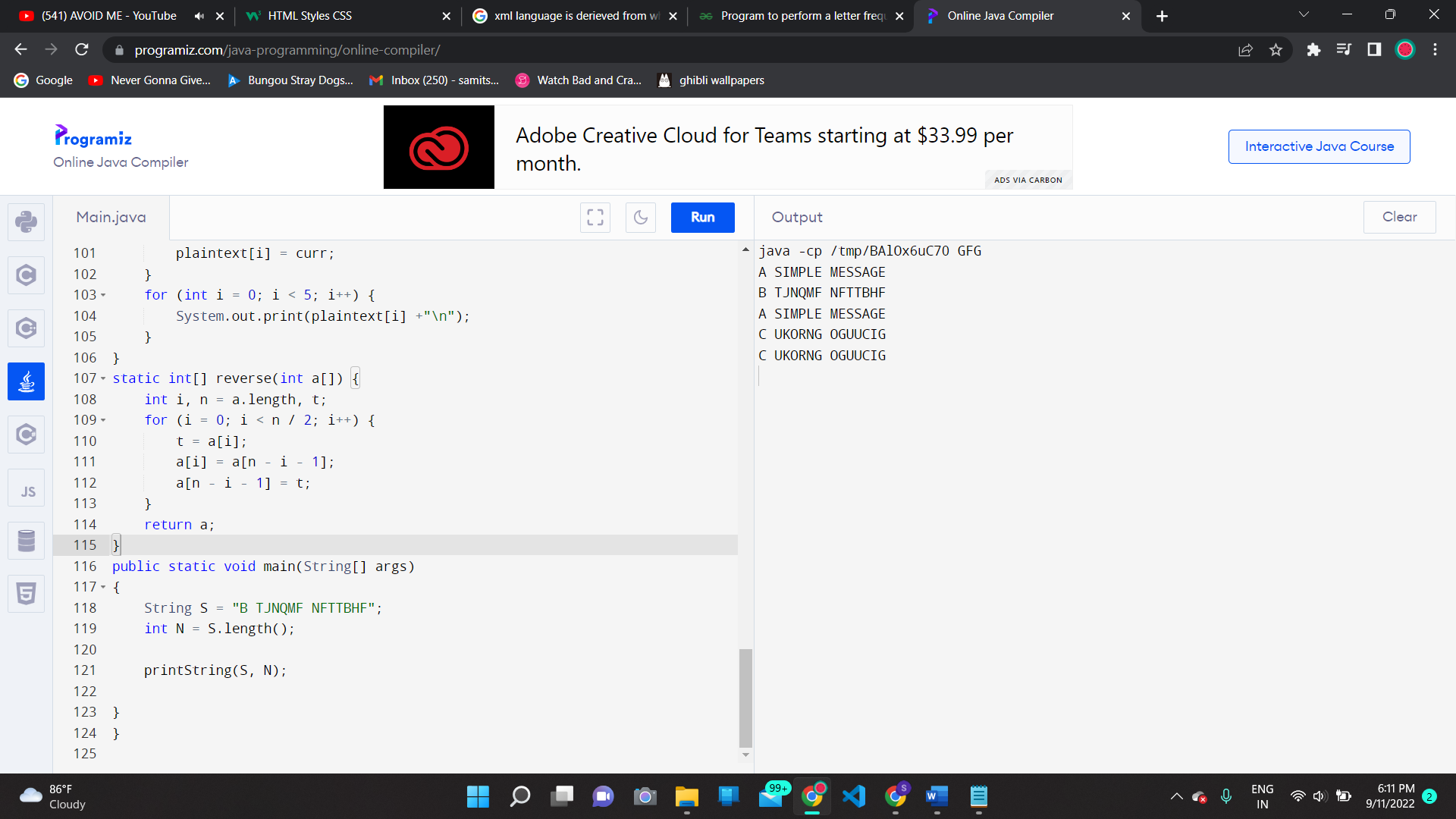
int N = S.length();

printString(S, N);

}

}

**Output:**



**Conclusion:**

Hence we have performed the experiment with frequency analysis method and got the desired output.

**TE-IT-B VPPCoE Page No: 03**