**Final Year B. Tech., Sem VI 2022-23**

**cryptography and network security Lab**

**PRN No: 2019BTECS00071**

**Name: Charpale Abhishek Parmanand**

**Batch: B3**

**Assignment 2 : cryptanalysis**

**Aim:**

To develop and implement the Cryptanalysis of Caesar Cipher and to test the code on the cipher input text

**Theory:**

* Cryptanalysis is the art of breaking codes and ciphers. The Caesar cipher is probably the easiest of all ciphers to break.
* Since the shift has to be a number between 1 and 25, (0 or 26 would result in an unchanged plaintext) we can simply try each possibility and see which one results in a piece of readable text.
* If you happen to know what a piece of the ciphertext is, or you can guess a piece, then this will allow you to immediately find the key.
* If this is not possible, a more systematic approach is to calculate the frequency distribution of the letters in the cipher text.
* This consists of counting how many times each letter appears. Natural English text has a very distinct distribution that can be used help crack codes.

**Code :**

#include <bits/stdc++.h>

using namespace std;

// https://cryptii.com/pipes/caesar-cipher

set<string> dict;

int main()

{

dict.insert("the");

dict.insert("gun");

dict.insert("is");

dict.insert("in");

dict.insert("the");

dict.insert("drawer");

string s, org;

cout << "Enter Cipher text" << endl;

getline(cin, s);

string x;

// for (int i = 0; i < s.length(); i++)

// if (s[i] != ' ')

// x += s[i];

// s = x;

int k = 0;

cout << "\nCipher text is: " << s << endl << endl;

org = s;

for (int k = 0; k < 26; k++)

{

cout << "Keep Key as: " << k << endl;

s = org;

string word = "";

int flg = 0;

for (int i = 0; i < s.length(); i++)

{

if (s[i] == ' ')

{

if (dict.find(word) == dict.end())

{

flg = 1;

break;

}

word = "";

continue;

}

int val = s[i] - 'a';

val = (val - k + 26) % 26;

char ch = 'a' + val;

word += ch;

s[i] = ch;

}

if (dict.find(word) == dict.end())

{

flg = 1;

}

if (flg == 0)

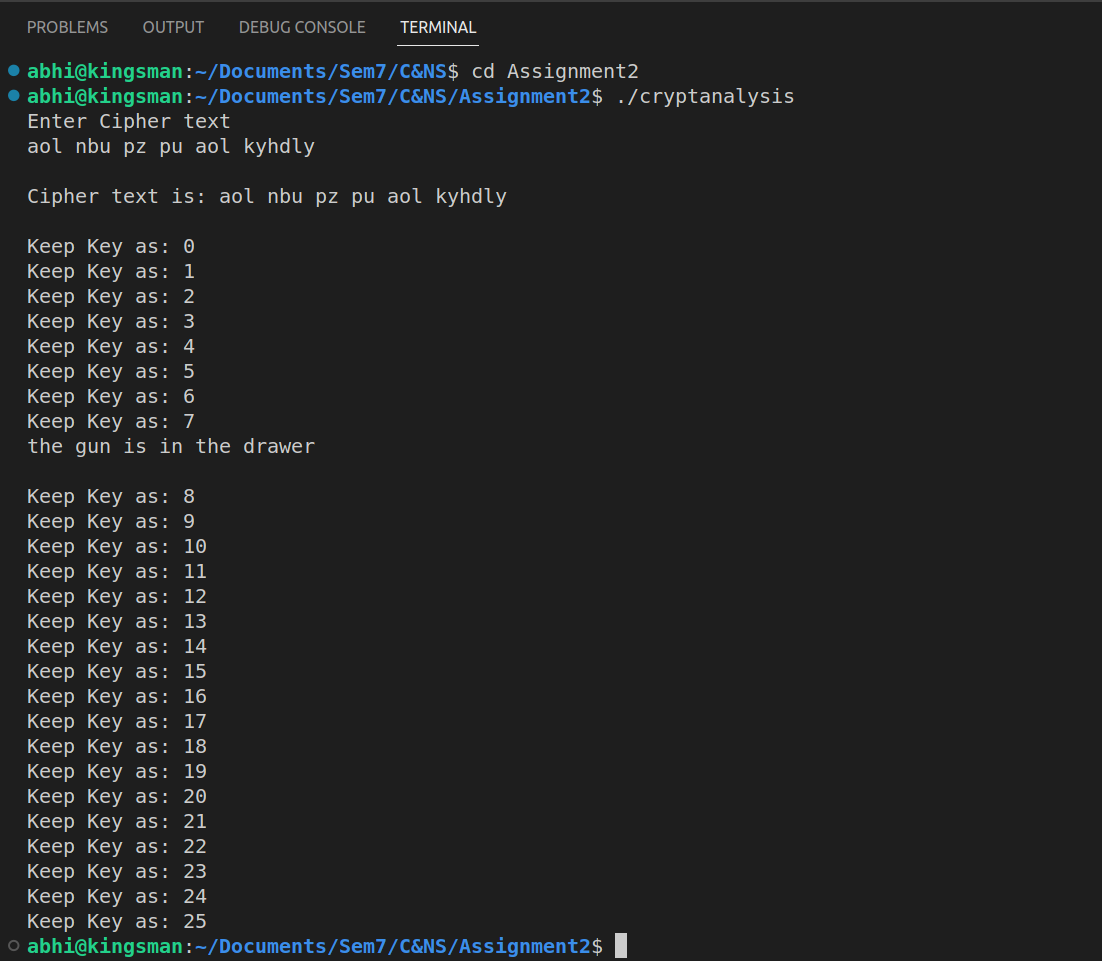
cout << s << endl << endl;

}

return 0;

}

**Output:**



**Conclusion**:

The code executed successfully and extracted the exact plain text from the ciphertext given by the input without key.