**Assignment No 4**

**Cryptography and Network Security Lab (5CS453)**

**Name: Sumit Narake PRN: 2020BTECS00023**

**Class: Final Year - CSE**

**Problem Statement:**

**Vigenère Cipher Technique**

**🡪**

The Vigenère Cipher Technique is a polyalphabetic substitution cipher that adds an extra layer of complexity to encryption by using a keyword or key phrase to determine the shifts applied to the plaintext letters.

Unlike monoalphabetic ciphers, where each letter is replaced with a fixed substitution, the Vigenère Cipher employs multiple alphabets with different shifts, making it more secure against frequency analysis.

**Key Setup:**

* Choose a keyword or key phrase.
* Replicate the keyword to match the length of the plaintext, repeating it as needed.
* Convert the keyword letters to their corresponding numerical values (A=0, B=1, ..., Z=25).

**Encryption:**

* Divide the plaintext into individual letters and convert them to numerical values.
* For each letter, determine the shift value using the corresponding keyword letter.
* Shift the plaintext letter by the calculated shift value (mod 26).
* Convert the shifted numerical value back to a letter to create the ciphertext.

**Decryption:**

* Divide the ciphertext into individual letters and convert them to numerical values.
* For each letter, determine the shift value using the corresponding keyword letter.
* Reverse the shift (subtract the shift value, mod 26).
* Convert the shifted numerical value back to a letter to retrieve the original plaintext.

**Advantages:**

* Stronger security due to polyalphabetic nature and keyword-driven shifts.
* Reduces susceptibility to frequency analysis.
* Key space increases with keyword length, enhancing security.

**Disadvantages:**

* Vulnerable to key length repetition (Kasiski examination) for shorter keywords.
* Security can weaken if the keyword is short or predictable.

**Code:**

#include<bits/stdc++.h>

using namespace std;

int main()

{

    int choice;

    cout << "Choose an option:\n";

    cout << "1. Encryption\n";

    cout << "2. Decryption\n";

    cout << "Enter your choice (1 or 2): ";

    cin >> choice;

    cin.ignore();  // Clear the newline character from the input buffer

    if (choice == 1)

    {

        // Encryption

        string plainText, key, cipherText;

        cout << "\nEnter plain text: ";

        getline(cin, plainText);

        cout << "\nEnter key: ";

        getline(cin, key);

        // Removing spaces and converting to lowercase from plaintext

        string temp = "";

        for (int i = 0; i < plainText.size(); i++)

        {

            if (plainText[i] != ' ')

                temp += plainText[i];

        }

        plainText = temp;

        for (int i = 0; i < plainText.size(); i++)

        {

            if (plainText[i] >= 'A' && plainText[i] <= 'Z')

                plainText[i] += 32; // Convert to lowercase

        }

        // Removing spaces and converting to lowercase from key

        string temp2 = "";

        for (int i = 0; i < key.size(); i++)

        {

            if (key[i] != ' ')

                temp2 += key[i];

        }

        key = temp2;

        for (int i = 0; i < key.size(); i++)

        {

            if (key[i] >= 'A' && key[i] <= 'Z')

                key[i] += 32; // Convert to lowercase

        }

        // Encryption

        for (int i = 0; i < plainText.size(); i++)

        {

            int val = plainText[i] - 'a' + key[i % key.size()] - 'a';

            cipherText += 'a' + (val % 26);

        }

        cout << "\nCipher Text: " << cipherText << endl;

    }

    else if (choice == 2)

    {

        // Decryption

        string cipherText, key;

        cout << "\nEnter cipher text: ";

        getline(cin, cipherText);

        cout << "\nEnter key: ";

        getline(cin, key);

        // Removing spaces and converting to lowercase from key

        string temp2 = "";

        for (int i = 0; i < key.size(); i++)

        {

            if (key[i] != ' ')

                temp2 += key[i];

        }

        key = temp2;

        for (int i = 0; i < key.size(); i++)

        {

            if (key[i] >= 'A' && key[i] <= 'Z')

                key[i] += 32; // Convert to lowercase

        }

        // Decryption

        string decrypted = "";

        for (int i = 0; i < cipherText.size(); i++)

        {

            int val = cipherText[i] - 'a' - (key[i % key.size()] - 'a') + 26;

            decrypted += 'a' + (val % 26);

        }

        cout << "\nAfter decryption: " << decrypted << endl;

    }

    else

    {

        cout << "Invalid choice. Please choose 1 or 2." << endl;

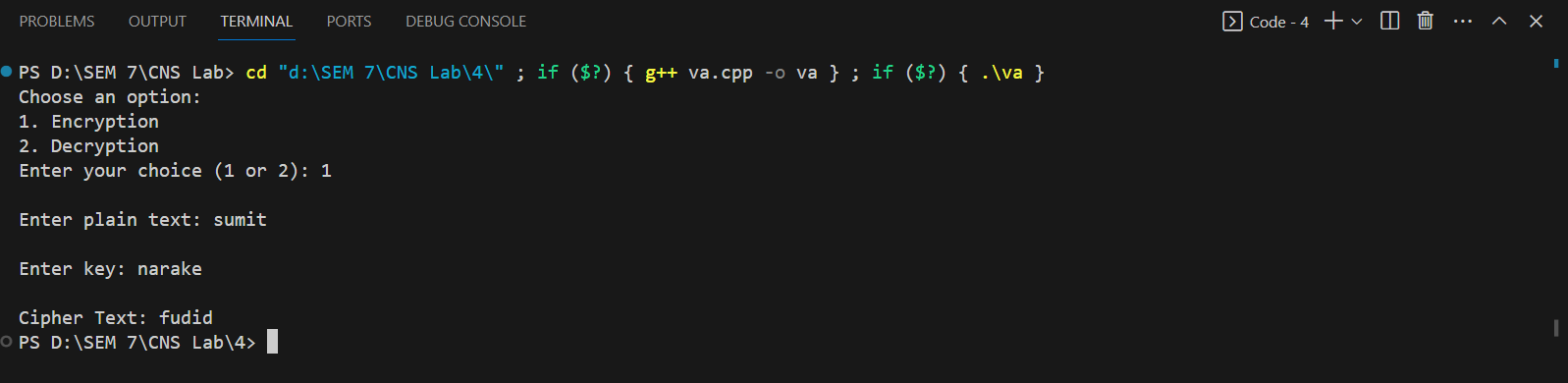
    }

    return 0;

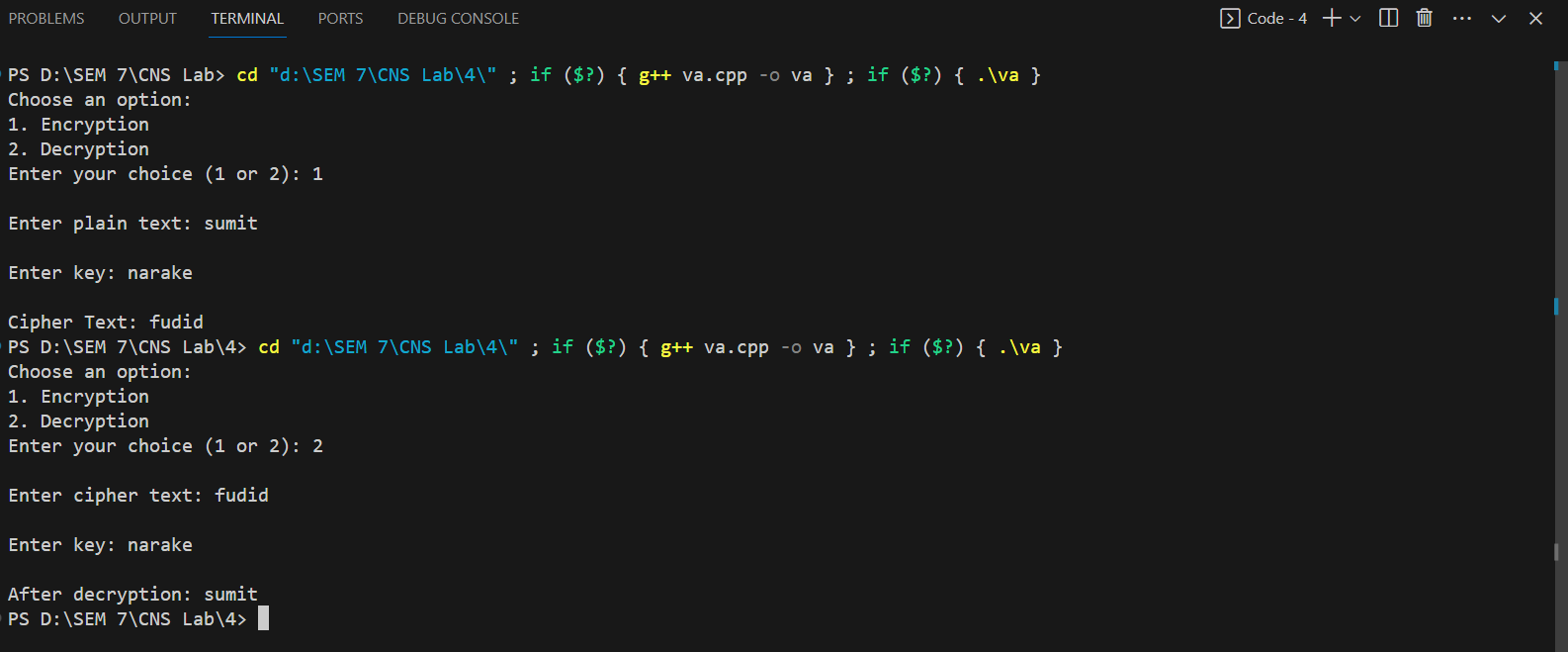
}

**Output:**

**Encryption:**

****

**Decryption:**

****