**Cryptography and Network Security Lab**

**Name: Rushikesh R. Ware**

**PRN: 2020BTECS00012**

**Batch: B2**

**VIGENERE ALGORITHM**

**Aim:**

To encrypt plain text using vigenere cipher and convert cipher text into plain text by decryption.

**Theory:**

Vigenere Cipher is a method of encrypting alphabetic text. It uses a simple form of polyalphabetic substitution. A polyalphabetic cipher is any cipher based on substitution, using multiple substitution alphabets. The encryption of the original text is done using the Vigenère square or Vigenère table.

**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;

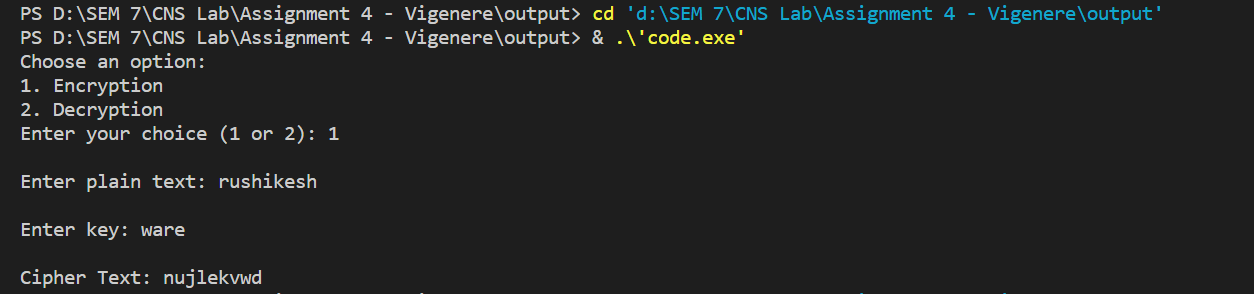
    }

    return 0;

}

**Output:**

**Encryption:**



**Decryption:**

