**Cryptography & Network Security Lab**

**Assignment 03**

**Vigenere Cipher:**

The Vigenère cipher is a method of encrypting alphabetic text by using a simple form of polyalphabetic substitution. Here are the steps to understand how it works:

1. Key Preparation: Choose a keyword or key phrase. This key should be as long as the message you want to encrypt. For example, if your key is "KEY," and your message is "HELLO," repeat the key to match the length of the message: "KEYKEY."

2. Alphabet Table: Create a table, often called a Vigenère Square, which consists of the alphabet repeated multiple times, shifting each row one position to the right. Here's an example of the first few rows of the table:

```

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

B C D E F G H I J K L M N O P Q R S T U V W X Y Z A

C D E F G H I J K L M N O P Q R S T U V W X Y Z A B

```

3. Encryption: Now, let's encrypt the message "HELLO" with the key "KEYKEY." Match each letter of the message with the corresponding letter of the key:

- "H" from the message is paired with "K" from the key, so the encrypted letter is "K."

- "E" from the message is paired with "E" from the key, so the encrypted letter is "E."

- "L" from the message is paired with "Y" from the key, so the encrypted letter is "Y."

- "L" from the message is paired with "K" from the key, so the encrypted letter is "K."

- "O" from the message is paired with "E" from the key, so the encrypted letter is "E."

The encrypted message is "KEYKE."

4. Decryption: To decrypt the message, you reverse the process. Use the same key to match each letter of the encrypted message with the corresponding letter of the key and find the corresponding letter in the Vigenère Square.

- "K" from the encrypted message is paired with "K" from the key, so the decrypted letter is "H."

- "E" from the encrypted message is paired with "E" from the key, so the decrypted letter is "E."

- "Y" from the encrypted message is paired with "Y" from the key, so the decrypted letter is "L."

- "K" from the encrypted message is paired with "K" from the key, so the decrypted letter is "L."

- "E" from the encrypted message is paired with "E" from the key, so the decrypted letter is "O."

The decrypted message is "HELLO."

This process continues for the entire message. The Vigenère cipher is more secure than a simple Caesar cipher because it uses a keyword that changes the substitution rules for each letter in the message, making it harder to decipher without the key.

#include <bits/stdc++.h>

using namespace std;

string format(string &*str*) {

    stringstream res;

    for (auto &ch : str) {

        if (ch != ' ') {

            res << (char)tolower(ch);

        }

    }

    return res.str();

}

string encrypt(string &*plain*, string &*key*) {

    stringstream cipher;

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

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

        cipher << (char)('a' + (val % 26));

    }

    return cipher.str();

}

string decrypt(string &*cipher*, string &*key*) {

    stringstream plain;

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

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

        plain << (char)('a' + (val + 26) % 26);

    }

    return plain.str();

}

int main() {

    int choice;

    cout << "1. Encrypt\n2. Decrypt\nEnter your choice: ";

    cin >> choice;

    cin.get();

    if (choice == 1) {

        string plain, key;

        cout << "\nEnter plain text: ";

        getline(cin, plain);

        plain = format(plain);

        cout << "\nEnter key: ";

        getline(cin, key);

        string cipher = encrypt(plain, key);

        cout << "\nEncrypted text is : " << cipher << endl;

    } else if (choice == 2) {

        string cipher, key;

        cout << "\nEnter cipher text: ";

        getline(cin, cipher);

        cipher = format(cipher);

        cout << "\nEnter key: ";

        getline(cin, key);

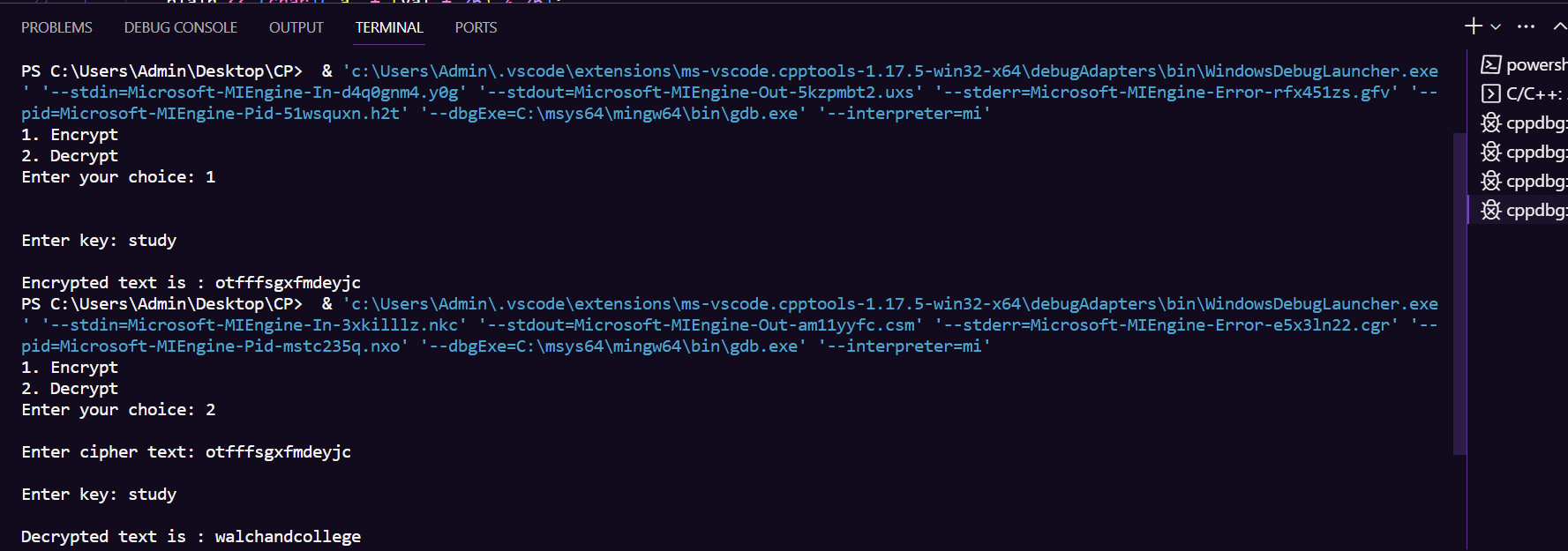
        string plain = decrypt(cipher, key);

        cout << "\nDecrypted text is : " << plain << endl;

    }

    return 0;

}



**Analysis:**

1. Encryption: The Row Transposition Cipher successfully encrypted the input plaintext "walchandcollege" using the key "study" to produce the ciphertext "otfffsgxfmdeyjc."

2. Decryption: When the same key "study" was used during decryption, the original plaintext "walchandcollege" was correctly recovered from the ciphertext "otfffsgxfmdeyjc."

The Row Transposition Cipher is a basic transposition cipher that rearranges the characters of a message based on a key. Here are some key points to consider:

1. Key Sensitivity: The order in which the rows are read during encryption and decryption is determined by the key. Using a different key would result in a different ciphertext and potentially produce gibberish during decryption.

2. Security Limitations: The Row Transposition Cipher provides minimal security and is relatively easy to break using modern cryptanalysis techniques. The security of the cipher primarily depends on the secrecy of the key, which, if discovered, can easily decrypt the message.

3. Non-Alphabetic Characters: This specific implementation of the Row Transposition Cipher appears to handle non-alphabetic characters without modification.

4. Padding: In your example, there was no need for padding characters because the plaintext length perfectly fit the matrix formed by the key "study." Padding characters are typically used when the plaintext length doesn't evenly divide by the number of rows specified in the key.

5. Limited Applicability: The Row Transposition Cipher is not suitable for encrypting large or complex messages and is typically more of a historical curiosity or a simple puzzle rather than a secure encryption method.

In summary, while the Row Transposition Cipher can successfully encrypt and decrypt messages, it is not recommended for securing sensitive information due to its lack of security. Modern cryptographic methods offer significantly stronger security for encryption purposes.