Cryptography & Network Security

PRN - 2019BTECS00026
Name - Niraja Vasudev Kulkarni
Batch - B1

Assignment - 4

Vigenere Cipher

- **Objective -**

  To implement Vigenere Cipher in C

- **Theory -**

Vigenere Cipher is 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 Vigenere table. The table consists of the alphabets written out 26 times in different rows, each alphabet shifted cyclically to the left compared to the previous alphabet, corresponding to the 26 possible Caesar Ciphers. At different points in the encryption process, the cipher uses a different alphabet from one of the rows. The alphabet used at each point depends on a repeating keyword.

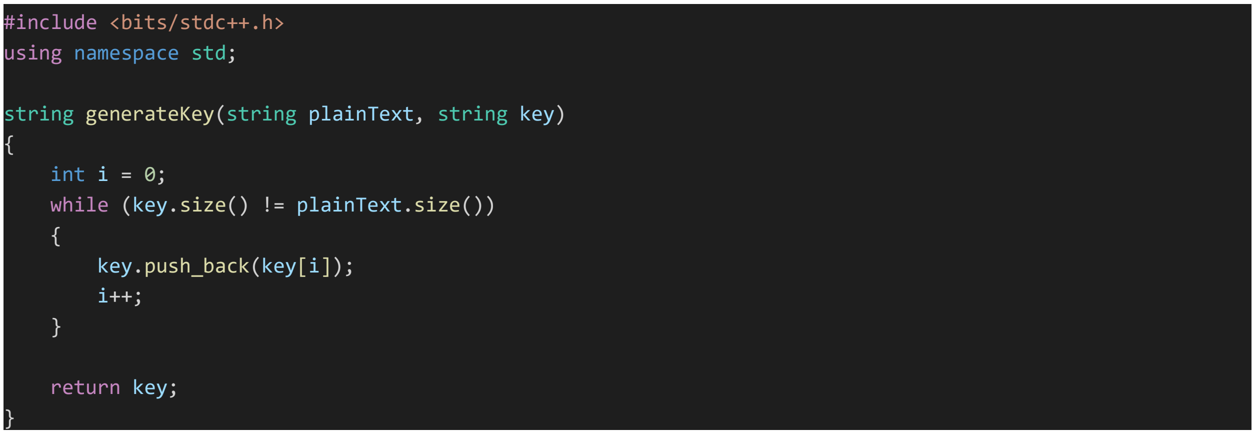
Procedure -

1. Take the input - plaintext & the key from user

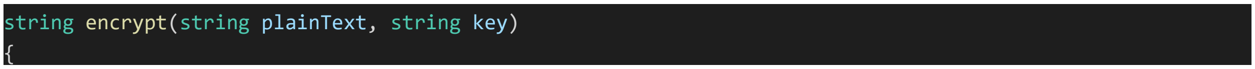
2. Make the key size same as text size by repeating it as required

3. Add each character of key & text and find out the cipher text accordingly

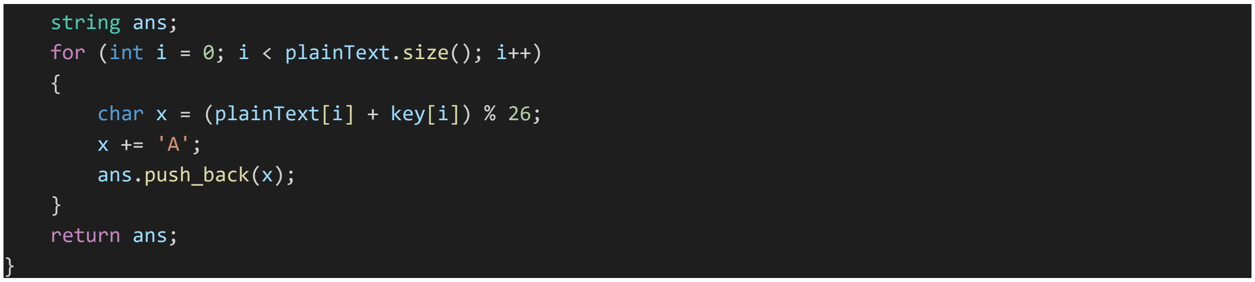
- **Code Snapshots -**



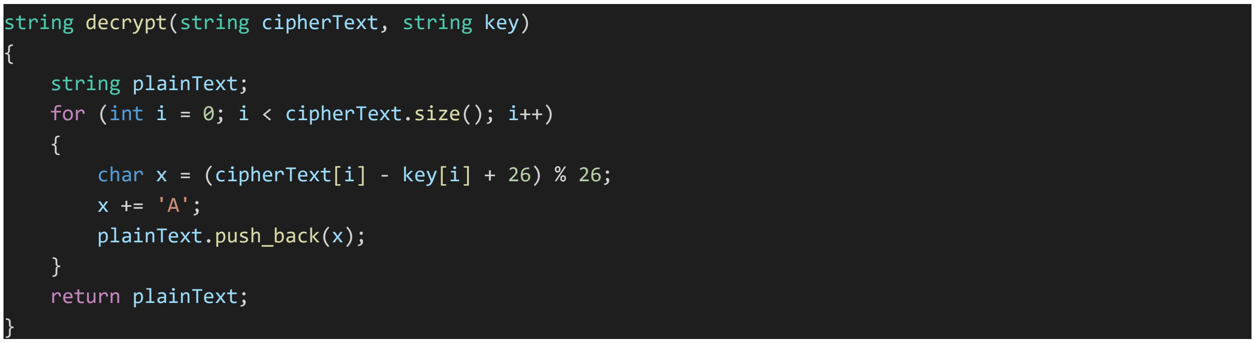
```
#include <bits/stdc++.h>
using namespace std;

string generateKey(string plainText, string key)
{
    int i = 0;
    while (key.size() != plainText.size())
    {
        key.push_back(key[i]);
        i++;
    }

    return key;
}
```



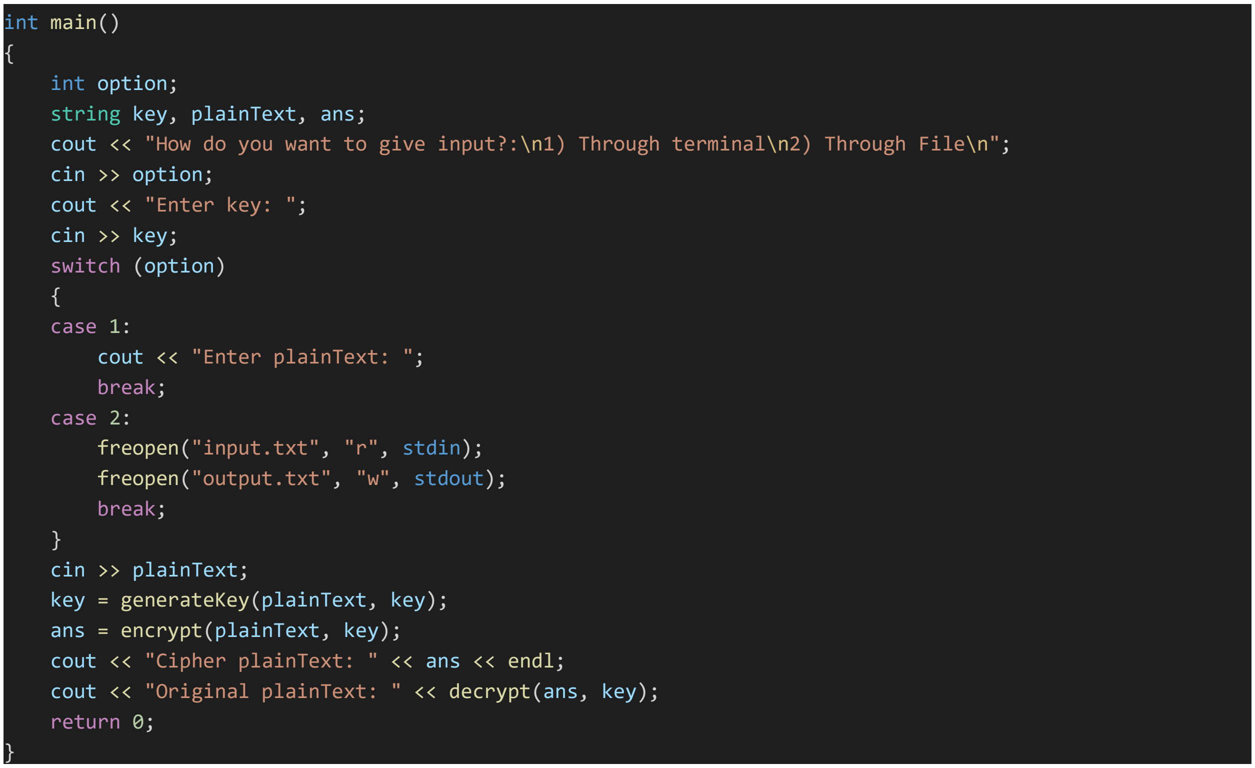
```
string encrypt(string plainText, string key)
{
```



```
    string ans;
    for (int i = 0; i < plainText.size(); i++)
    {
        char x = (plainText[i] + key[i]) % 26;
        x += 'A';
        ans.push_back(x);
    }
    return ans;
}
```

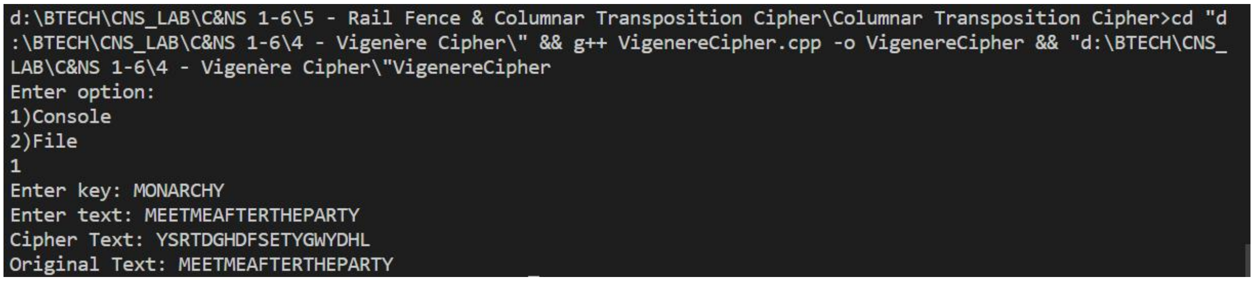


```
string decrypt(string cipherText, string key)
{
    string plainText;
    for (int i = 0; i < cipherText.size(); i++)
    {
        char x = (cipherText[i] - key[i] + 26) % 26;
        x += 'A';
        plainText.push_back(x);
    }
    return plainText;
}
```



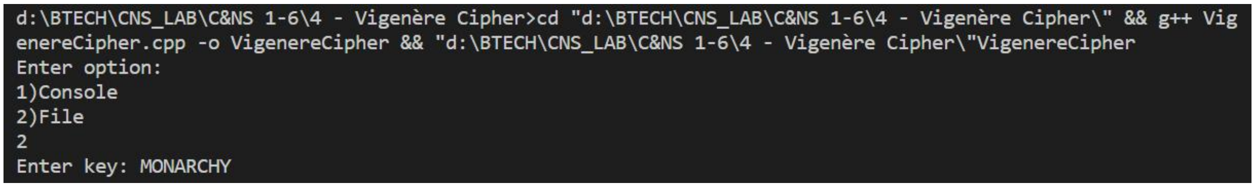
```
int main()
{
    int option;
    string key, plainText, ans;
    cout << "How do you want to give input?:\n1) Through terminal\n2) Through File\n";
    cin >> option;
    cout << "Enter key: ";
    cin >> key;
    switch (option)
    {
    case 1:
        cout << "Enter plainText: ";
        break;
    case 2:
        freopen("input.txt", "r", stdin);
        freopen("output.txt", "w", stdout);
        break;
    }
    cin >> plainText;
    key = generateKey(plainText, key);
    ans = encrypt(plainText, key);
    cout << "Cipher plainText: " << ans << endl;
    cout << "Original plainText: " << decrypt(ans, key);
    return 0;
}
```

- **Outputs -**
- **Sample Output 1 -**



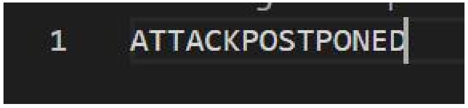
```
d:\BTECH\CNS_LAB\C&NS 1-6\5 - Rail Fence & Columnar Transposition Cipher\Columnar Transposition Cipher>cd "d
:\BTECH\CNS_LAB\C&NS 1-6\4 - Vigenère Cipher\" && g++ VigenereCipher.cpp -o VigenereCipher && "d:\BTECH\CNS_
LAB\C&NS 1-6\4 - Vigenère Cipher\"VigenereCipher
Enter option:
1)Console
2)File
1
Enter key: MONARCHY
Enter text: MEETMEAFTERTHEPARTY
Cipher Text: YSRTDGHDFSETYGWYDHL
Original Text: MEETMEAFTERTHEPARTY
```

- **Sample Output 2-**



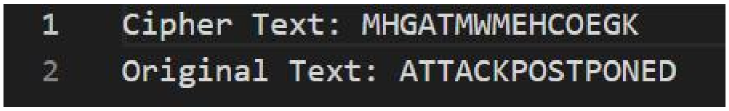
```
d:\BTECH\CNS_LAB\C&NS 1-6\4 - Vigenère Cipher>cd "d:\BTECH\CNS_LAB\C&NS 1-6\4 - Vigenère Cipher\" && g++ Vig
enereCipher.cpp -o VigenereCipher && "d:\BTECH\CNS_LAB\C&NS 1-6\4 - Vigenère Cipher\"VigenereCipher
Enter option:
1)Console
2)File
2
Enter key: MONARCHY
```

Input file -



```
ATTACKPOSTPONED
```

Output file -



```
Cipher Text: MHGATMWMEHCOEGK
Original Text: ATTACKPOSTPONED
```

- **Conclusion -**

Vigenere Cipher is polyalphabetic substitution cipher , in which a single alphabet can be encrypted with different alphabets when its occurrence is repeated.