Assignment 4 : Vigenere Encryption and Decryption

PRN No: 2020BTECS00057

Name: Mrunal Khade

Batch: B4

Aim:

To develop and implement the Vigenere Cipher and to encryption and decryption on the input plaintext

Theory:

- Vigenere Cipher is a method of encrypting alphabetic text. It uses a simple form of polyalphabetic substitution.
- The encryption of the original text is done using the Vigenere square or 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.

Code :

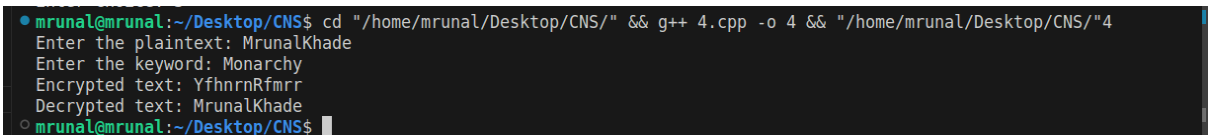
```
#include<bits/stdc++.h>

using namespace std;

class Vigenere

{

   public:

       string key;

   void createkey(string k) {

       key.clear();

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

       {

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

               key += k[i];

           else if (k[i] >= 'a' && k[i] <= 'z')

               key += k[i] + 'A' - 'a';

       }

   }

   string encryption(string t)

   {

       string output;

       for (int i = 0, j = 0; i < t.length(); ++i)

       {

           char c = t[i];
```

```
            if(c == ' ')
                continue;
            if (c >= 'a' && c <= 'z')
                c += 'A' - 'a';
            else if (c < 'A' || c > 'Z')
                continue;
            output += (c + key[j] - 2 * 'A') % 26 + 'A';
            //added 'A' to bring it in range of ASCII alphabet [
65-90 | A-Z ]
            j = (j + 1) % key.length();
        }
        return output;
    }
    string decryption(string t)
    {
        string output;
        for (int i = 0, j = 0; i < t.length(); ++i)
        {
            char c = t[i];
            if (c >= 'a' && c <= 'z')
                c += 'A' - 'a';
            else if (c < 'A' || c > 'Z')
                continue;
            output += (c - key[j] + 26) % 26 + 'A';
```

```
            //added 'A' to bring it in range of ASCII alphabet [
65-90 | A-Z ]

            j = (j + 1) % key.length();

        }

        return output;

    }

};



int main()

{

    Vigenere v;

    int choice;

    int datachoice;

    string sample,key;

    int shift;

    while(1)

    {

        cout << "Vigenere Cipher\n 1. Encryption \n 2. Decryption\n
3. Exit\nEnter Choice: ";

        cin>>choice;

        if(choice>2)

            break;

        switch(choice)

        {

            case 1:
```

```
                cout<<"Enter data to be Encrypted:\n";
                cin.ignore();
                getline(cin,sample);
                cout<<"Enter the key: ";
                getline(cin,key);
                v.createkey(key);
                cout<<"Encrypted String:\n";
                cout<<v.encryption(sample)<<endl;
                break;
            case 2:
                cout<<"Enter data to be Decrypted:\n";
                cin.ignore();
                getline(cin,sample);
                cout<<"Enter the key: ";
                getline(cin,key);
                v.createkey(key);
                cout<<"Decrypted String:\n";
                cout<<v.decryption(sample)<<endl;;
                break;
        }
    }
    return 0;
}
```

Output:



Conclusion:

Performed the experiment successfully. Encrypted the data with the provided key. Output of this encryption is decrypted to match the plaintext that was inputted by the user as shown in the above diagram.