Politechnika Śląska Wydział Automatyki, Elektroniki i Informatyki

Podstawy Programowania Komputerów

TEMAT PROJEKTU: **VIGENERE**

Autor: Kamil Niedziela

Prowadzący: mgr inż. Dariusz Marek

Rok Akademicki: 2020/2021

Kierunek: Informatyka

Rodzaj studiów: SSI

Termin oddania sprawozdania: 2020-12-15

1. Treść zadania

22 Vigenère

Proszę napisać program, który:

1. szyfruje pliki tekstowe metodą Vigenère'a:

   Program uruchamiany jest z linii poleceń z wykorzystaniem następujących przełączników (kolejność przełączników jest dowolna):

–en	flaga szyfrowania
-i	plik tekstowy wejściowy jawny
-o	plik tekstowy wyjściowy zaszyfrowany
-k	plik tekstowy z kluczem
-h	wyświetlenie wszystkich możliwych przełączników oraz krótką instrukcję.

   przykładowo: **./main –en -i jawny -k klucz -o zaszyfrowany**

2. deszyfruje pliki tekstowe zaszyfrowane metodą Vigenère'a:

–de	flaga szyfrowania
-i	plik tekstowy wyjściowy zaszyfrowany
-o	plik tekstowy wejściowy jawny
-k	plik tekstowy z kluczem
-h	wyświetlenie wszystkich możliwych przełączników oraz krótką instrukcję.

   przykładowo: **./main –de -k klucz -o odszyfrowany -i tajny**

3. łamie pliki tekstowe zaszyfrowane metodą Vigenère'a:

–br	flaga łamania szyfru
-i	plik tekstowy wejściowy zaszyfrowany
-o	plik tekstowy wyjściowy jawny
-h	wyświetlenie wszystkich możliwych przełączników oraz krótką instrukcję

   przykładowo: **./main –br -o odszyfrowany -i tajny -k klucz**

Uruchomienie programu bez parametrów powoduje wypisanie tej samej instrukcji, co w przypadku uruchomienia programu z przełącznikiem *-h*.

2. Analiza zadania

Zrealizowany przeze mnie program umożliwia szyfrowanie lub deszyfrowanie metodą Vigenere oraz łamanie klucza szyfrującego na podstawie danych w plikach tekstowych.

2.1. Struktury danych

Program oparty jest na wektorach które odpowiadają za funkcję bufora tekstowego lub gromadzą znaki.

2.2. Algorytmy

W programie obecnych jest kilka algorytmów, działają w podobny sposób. Szyfrowanie oraz deszyfrowanie wykonywane jest na podstawie przesunięć znaków w tabeli ASCII. W przypadku gdy przesuwający klucz szyfrujący przesunie litery wyjściowe poza zakres liter alfabetu łacińskiego, program szuka litery cyklicznie od początku lub końca alfabetu. Algorytm odpowiadający za łamanie klucza szyfrującego wykonywany jest na podstawie pliku wejściowego, wyjściowego oraz znajomości położenia znaków w tabeli ASCII. Odejmując odpowiednio kody znaków wejściowych od znaków wyjściowych program potrafi określić znaki klucza szyfrującego.

3. Specyfikacja zewnętrzna

Program uruchamiany jest z wiersza poleceń. Do programu należy podać kilka parametrów, są to: plik z rozszerzeniem .exe, flaga szyfrująca, deszyfrująca lub łamiąca klucz w zależności od intencji użytkownika, plik wejściowy po przełączniku „-i”, plik wyjściowy po przełączniku „-o”, plik z kluczem szyfrującym po przełączniku „-k”. Kolejność przełączników jest dowolna.

Wzór na wprowadzenie danych do wiersza poleceń dla mojego programu:

ScieżkaDoPliku>Vigenere.exe -en -i plikWejsciowy.txt -o plikWyjsciowy.txt -k kluczSzyfrujacy.txt

ScieżkaDoPliku>Vigenere.exe -de -i plikWejsciowy.txt -o plikWyjsciowy.txt -k kluczSzyfrujacy.txt

ScieżkaDoPliku>Vigenere.exe -br -i plikWejsciowy.txt -o plikWyjsciowy.txt -k kluczSzyfrujacy.txt

Na podstawie odpowiedniej flagi program wykonuje kolejne operacje na danych.

Uruchomienie programu z większą ilością flag niż jedna wyświetli odpowiedni komunikat, podobnie jak wprowadzanie za małej ilości przełączników, brak wprowadzenia nazwy pliku po przełączniku lub braku podania rozszerzenia .txt po nazwach plików.

4. Specyfikacja wewnętrzna

Program został zrealizowany zgodnie z paradygmatem strukturalnym.

4.1. Ogólna struktura programu

W funkcji głównej *main* program wczytuje przełączniki pobrane przez wiersz poleceń. Odpowiada za to funkcja *loadSwitches.* Następnie wczytuje dane z pliku wejściowego do bufora za pomocą funkcji *loadDataToBuffer*. Potem zostaje wywołana funkcja *vigenere* która odpowiada za wszystkie operacje(szyfrowanie, deszyfrowanie i łamanie klucza) na danych. Ostatnim etapem aplikacji jest wywołanie funkcji *saveToFile.* Funkcja ta odpowiada za zapis danych do pliku. Pomyślne wywołanie programu skutkuje wyświetleniem komunikatu *" Program zostal wykonany pomyslnie, zamykanie programu."* W przypadku błędnego wczytania przełączników program zakończy się wyświetlając komunikat *„Nie udalo sie poprawnie wczytac przelacznikow, zamykanie programu.".*

5. Testowanie

Aplikacja została przetestowana względem błędnych danych wejściowych.

6. Wnioski

Program który stworzyłem wymaga wymyślenia kilku algorytmów odpowiedzialnych za szyfrowanie, deszyfrowanie oraz łamanie klucza szyfrującego. Szczególną trudnością okazało się dla mnie stworzenie funkcji usuwającej powtarzające się podsłowa w algorytmie szukającym klucza szyfrującego. Mimo wszystko poradziłem sobie z problemem czego wynikiem jest gotowy program.

Literatura i źródła:

Język C++. Szkoła programowania. Wydanie VI. Autor: Stephen Prata

https://stackoverflow.com/