Белорусский государственный технологический университет

Факультет информационных технологий

Кафедра программной инженерии

Специальность: Программная инженерия

Группа: 10

Подгруппа: 1

Дисциплина: Основы алгоритмизации и программирования

 Реферат

На тему «Шифрование и дешифрование»

Выполнил:

Студент: Сегренёв Кирилл Сергеевич

Проверил:

Преподаватель: Белодед Н.И

2023, Минск

Содержание

[**ВВЕДЕНИЕ** 3](#_Toc150957930)

[**ПОСТОНОВКА ЗАДАЧИ** 4](#_Toc150957931)

[**АЛГОРИТМ** 5](#_Toc150957932)

[**КОДЫ ПРОГРАММ** 8](#_Toc150957933)

[**ВЫВОД** 10](#_Toc150957934)

**ВВЕДЕНИЕ**

На лекции 14.11 преподавателем была поставлена задача написать шифратор и дешифратор, которые запускаются через командную строку(берут данные из отдельного файла .txt) и записывают данные в новый txt-файл.

Главная задача стоит не в том, чтобы написать более сложный шифр, а в том, чтобы научиться брать из файлов данные и записывать их в новые файлы.

**ПОСТОНОВКА ЗАДАЧИ**

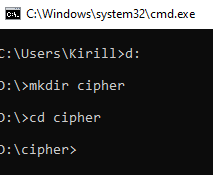
В данной задаче нам требуется написать шифратор и дешифратор, сделаю я это через 2 исполняемых файла. Мне придется писать код отдельно как для шифратора, так и для дешифратора.

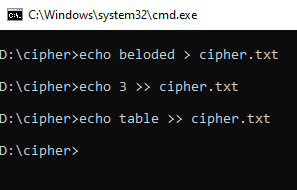
Я буду применять шифр Цезаря(сдвиг букв латинского алфавита на несколько позиций вперед) и после шифра цезаря будет применяться ключ-слово, которое будет дополнительно шифровать уже зашифрованное слово(прибавление кода символа ключевого слова к коду символа слова, которое нам нужно зашифровать). Все это будет происходить только в пределах латинского алфавита.

Так же дополнительно я хотел воспользоваться шифром ADFGX, но алгоритм написания кода мне показался сложным, но если вас заинтересует данный шифр, я могу его объяснить перед либо после лекции.

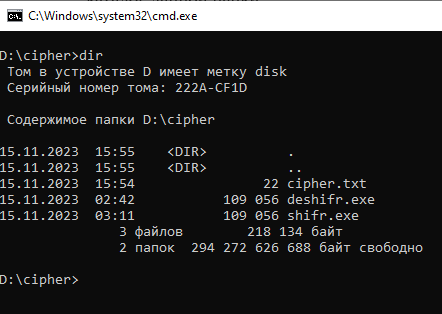
Зашифруем слово beloded с ключ-цифрой 3 и ключ-словом table

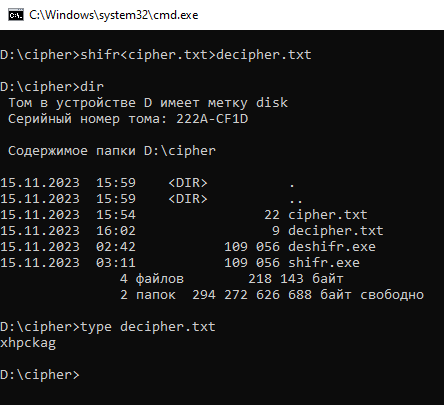
**АЛГОРИТМ**

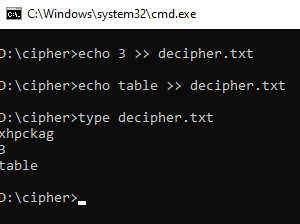
С помощью cmd создадим отдельную папку, где будут находится все файлы, относящиеся к данному заданию.  
  
С помощью d: переместимся в каталог диска D и там с помощью команды mkdir создадим папку cipher. Далее с помощью команды cd перейдем в каталог данной папки.

C помощью команды echo создаем файл с данными, которые нам нужно зашифровать и ключи, которые применяются при шифровании  


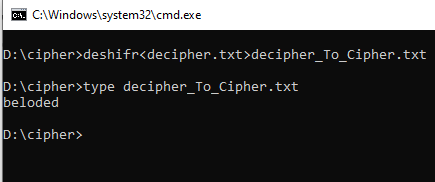
Переместим данные файлы из каталога VS в нашу папку и проверим, какие файлы находятся в данной папке с помощью dir

  
В папке cipher находится 3 файла, два исполняемых(.exe) и один текстовый(.txt)  
Далее нам нужно запустить исполняемый файл шифратора, взять данные из файла cipher.txt и записать их уже зашифрованными в новый txt-файл.

С помощью команд проверим, создался ли файл в данном каталоге и что в нем написалось.  


Допишем в decipher.txt наши ключи, которые использовались при шифровании для следующего дешифрования.  


Далее запустим через консоль deshifr.exe в который внесем данные из decipher.txt и создадим новый .txt файл c расшифрованным текстом.

   
  
Коды шифратора и дешифратора буду представлены в следующем пункте

**КОДЫ ПРОГРАММ**

**Код шифратора:**

#include <iostream>

#include <string>

using namespace std;

int main()

{

string wordCipher; // Слово, которое нам нужно зашифровать

int keyCaesar; // Число, на сколько должна сдвинуться буква в алфавите

string keyWord; // Ключ-слово, которое применяется во втором методе шифрования

cin >> wordCipher >> keyCaesar >> keyWord;

// Применение шифра Цезаря

for (auto& charCipher : wordCipher) { // В данном цикле мы объявляем переменную charCipher, которая инициализируется каждым символом в слове

// Шифрование для символов в верхнем регистре

if (charCipher >= 'A' && charCipher <= 'Z') {

charCipher = ((charCipher - 'A' + keyCaesar) % 26) + 'A';

}

// Шифрование для символов в нижнем регистре

if (charCipher >= 'a' && charCipher <= 'z') {

charCipher = ((charCipher - 'a' + keyCaesar) % 26) + 'a';

}

}

// Второе шифрование(с помощью ключа-слова)

char keyWordChar;

for (int i = 0; i < wordCipher.size(); ++i) { // С помощью функции size() получаем в int число букв в слове

// Получение символа ключа-слова

keyWordChar = keyWord[i % keyWord.size()]; //Используем взятие остатка от деления для того, чтобы, если ключ-слово меньше, чем то слово, которое нам нужно зашифровать, ключ-слово начиналось с начала(происходила своеобразная итерация слова)

// Шифрование для символов в верхнем регистре

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

wordCipher[i] = ((wordCipher[i] - 'A' + (keyWordChar - 'A')) % 26) + 'A'; // %26 используется для того, чтобы буквы по своему коду оставались в диапазоне от A до Z(циклический сдвиг в пределах алфавита с учетом регистра)

}// +'A' добавляем для корректного отображения символов в верхнем регистре

// Шифрование для символов в нижнем регистре

if (wordCipher[i] >= 'a' && wordCipher[i] <= 'z') {

wordCipher[i] = ((wordCipher[i] - 'a' + (keyWordChar - 'a'))%26) + 'a';

}

}

cout << wordCipher << endl;

return 0;

}

**Код дешифратора:**

#include <iostream>

#include <string>

using namespace std;

int main()

{

string wordCipher; // Слово, которое нам нужно расшифровать

int keyCaesar; // Ключ-число, на сколько шифр был сдвинут в шифровальщике

string keyWord; // Ключ-слово, которое применялось для зашифровки

cin >> wordCipher >> keyCaesar >> keyWord;

char keywordChar;

// Дешифрование с помощью ключа-слова, которое применялось в зашифровке слова

// Т.к последним действием в шифровании мы применяли ключ-слово, то первым действием в дешифровании мы должны мы должны сделать противоположное действие с данным ключ-слово

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

// Получение символа ключа-слова

keywordChar = keyWord[i % keyWord.size()];

// Дешифрование для символов в верхнем регистре

// Здесь мы отнимаем от каждого кода символа зашифрованного текста код символа ключа-слова

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

wordCipher[i] = ((wordCipher[i] - 'A' - (keywordChar - 'A') + 26) % 26) + 'A';

// + 26 мы используем, чтобы число, от которого мы берем остаток от деление, было положительным, что обеспечивает нам дешифрование в пределах алфавита, а не других символов

}

// Дешифрование для символов в нижнем регистре

if (wordCipher[i] >= 'a' && wordCipher[i] <= 'z') {

wordCipher[i] = ((wordCipher[i] - 'a' - (keywordChar - 'a') + 26) % 26) + 'a';

}

}

// Дешифрование шифра Цезаря (применяется ключ-число, с помощью которого было зашифровано слово в шифровальщике)

// Здесь же мы сдвигаем буквы на позиции назад, путем отнимания от их кода ключ-число(оставаясь в пределах алфавита)

for (auto& charCipher : wordCipher) {

if (charCipher >= 'A' && charCipher <= 'Z') {

charCipher = ((charCipher - 'A' - keyCaesar + 26) % 26) + 'A';

}

if (charCipher >= 'a' && charCipher <= 'z') {

charCipher = ((charCipher - 'a' - keyCaesar + 26) % 26) + 'a';

}

}

cout << wordCipher << endl;

return 0;

}

**ВЫВОД**

В данном реферате я применял шифрование и дешифрование слова beloded, слово полностью состоит из символов нижнего регистра, но в моем коде можно заметить, что так же можно посчитать символы и верхнего регистра.  
Если взять слово BeLoDeD данная программа так же сможет его зашифровать и расшифровать.

В данном коде я не применял функции open(), close(), а успользовал только ввод/вывод в консоле.

При создании файлов и папок (кроме исполняемых файлов) я пользовался cmd