Лабораторная работа № 4

Якупов Джамиль 4117

Требования:

Дружественный интерфейс.

2. Возможность многократного ввода исходных данных необходимых для решения поставленной задачи. Программа должна спрашивать: «Хотите повторить ввод исходных данных? Да — 1, Нет — 0.» Также в некотором виде должен формироваться запрос(ы), определяющие откуда поступят исходные данные и куда будет осуществлён вывод результата.

3. Использование минимум одной функции помимо функции main.

4. Ввод исходных данных из файла (путь к файлу задаётся в коде).

5. Ввод исходных данных из консоли.

6. Вывод результатов в файл (путь к файлу задаётся в коде).

7. Вывод результатов в консоль.

8. В случае ввода данных из файла программа должна завершаться (не предлагать повторно ввести исходные данные).

9. Использование uniform инициализации c++.

10. Защиту от некорректного пользовательского ввода. При этом следует продумать возможные случаи такого некорректного ввода. Перечислить в отчёте примеры возможных вариантов некорректного пользовательского ввода.

11. Размещение пользовательских констант в отдельном файле, например, "constants.h" с constants.h"constants.h" с с «header guards».

12. Размещение прототипов пользовательских функций в отдельном файла, например, "constants.h" с myfuncs.h"constants.h" с с «header guards». Допускается несколько таких файлов. 13. Размещение описаний пользовательских функций в отдельном файла, например, "constants.h" с myfuncs.cpp"constants.h" с . Допускается несколько таких файлов.

14. Передачу статических массивов в функции(ю) в качестве параметров.

15. Оформить по выполненной л.р. в тетради письменный отчёт2 .

16. Передачу статических массивов в функции(ю) в качестве параметров.

17. Динамическое выделение памяти.

18. Передачу динамических массивов в качестве параметров функций.

Работу со строками произвольной длины.

19. Ввод исходных данных из файла (путь к файлу задаётся с консоли, см п.4).

20. Вывод результатов в файл (Путь к файлу задаётся с консоли, если он не указывается, то используется путь к файлу по умолчанию. То есть, например программа выдаёт такой запрос: «Укажите файл для вывода результатов работы программы [~`rez`out.txt]:». Если просто нажать Enter, то файлом вывода будет файл «~`rez`out.txt»).

Задание:

269. Даны натуральное число п, символы S1, ..., Sn. Группы

символов, разделенные пробелами (одним или несколькими) и не

содержащие пробелов внутри себя, будем называть словами,

Подсчитать количество слов в данной последовательности.

Код программы:

Main.cpp

#include <iostream>

#include <string>

#include "myFuncs.h"

using namespace std;

int main()

{

setlocale(LC\_ALL, "Russian");

string line;

bool consoleInput{}, consoleOutput{};

do

{

consoleInput = inputMenu();

consoleOutput = outputMenu();

if (consoleInput)

{

cout << "введите строку" << endl;

cin.ignore();

getline(cin, line);

}

else

{

line = fileInput();

}

if (consoleOutput)

{

cout << answer2(line) << endl;

}

else

{

fileOutput(to\_string(answer2(line)));

}

if (consoleInput)

{

if (repeat() == false)

{

consoleInput = false;

break;

}

else

{

consoleInput = false;

}

}

else

{

break;

}

} while (true);

}

Myfuncs.cpp

#include "myfuncs.h"

#include "constants.h"

bool inputMenu()

{

int enter;

bool isConsoleInput = false ;

while (true)

{

cout << "Откуда взять данные?" << endl << "2 - С файла; 1 - с консоли" << endl;

cin >> enter;

if (!cin or (enter != 1 and enter != 2))

{

cout << "Неправильный ввод данных!\n";

cin.clear();

while (cin.get() != '\n');

}

else break;

}

if (enter == 1)

isConsoleInput = true;

return isConsoleInput;

}

bool outputMenu()

{

int enter;

bool isConsoleOutput{ false };

while (true)

{

cout << "Куда вывести?. 2 - файл, 1 - консоль" << endl;

cin >> enter;

if (!cin or (enter != 1 and enter != 2))

{

cout << "Неправильный ввод данных!\n";

cin.clear();

while (cin.get() != '\n');

}

else break;

}

if (enter == 1)

isConsoleOutput = true;

return isConsoleOutput;

}

string fileInput()

{

string path, line;

cout << "Введите путь к файлу. Если нажать Enter, то файлом ввода будет файл input.txt\n";

cin.ignore();

getline(cin, path);

if (path.length() == 0)

{

path = INPUT\_FILE\_NAME;

}

ifstream fIn(path);

getline(fIn, line);

return line;

}

void fileOutput(string answer)

{

string path;

cout << "Введите путь к файлу. Если нажать Enter, то файлом вывода будет файл output.txt\n";

getline(cin, path);

if (path.length() == 0)

{

path = OUTPUT\_FILE\_NAME;

}

ofstream fOut(path);

fOut << answer;

fOut.close();

}

int answer2(string c)

{

int res = 0;

for (int i = 1; i < c.size() - 1; i++)

{

if (c[i] == ' ' && c[i - 1] != ' ')

{

res += 1;

}

}

return res + 1;

}

double stringToNum(string str)

{

double num = atof(str.c\_str());

return num;

}

bool repeat()

{

int enter;

while (true)

{

cout << "Хотите еще, 1 - да, 2 - нет" << endl;

cin >> enter;

if (!cin or (enter != 1 and enter != 2))

{

cout << "Неправильный ввод данных!\n";

cin.clear();

while (cin.get() != '\n');

}

else break;

}

if (enter == 1)

{

return true;

}

else

{

return false;

}

}

Myfuncs.h

#pragma once

#include <iostream>

#include <fstream>

#include <string>

using namespace std;

bool inputMenu();

bool outputMenu();

string fileInput();

void fileOutput(string answer);

int answer2(string c);

string answer(string phrase);

bool repeat();

constants.h

#pragma once

#include <string>

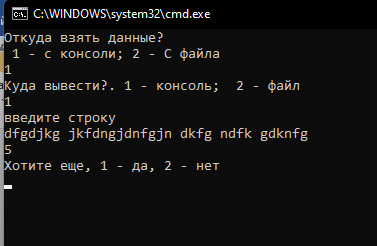
const std::string INPUT\_FILE\_NAME = "input.txt";

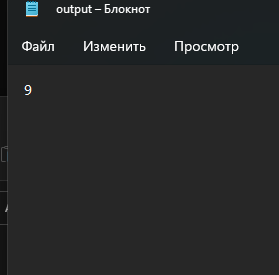
const std::string OUTPUT\_FILE\_NAME = "output.txt";

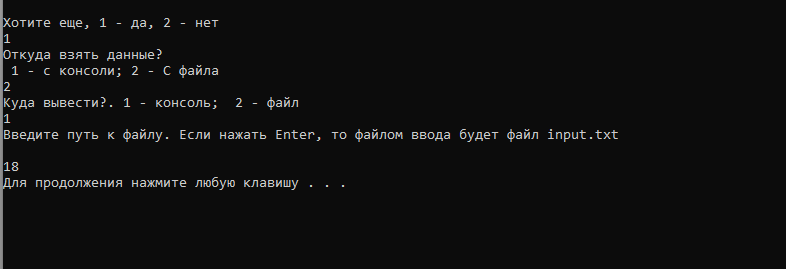
input.txt

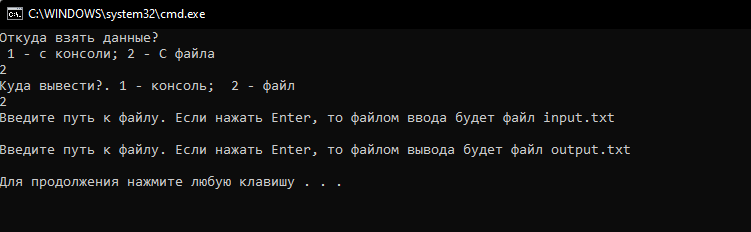
Мы пьем не потому, что тянемся к веселью, и не разнузданность себе мы ставим целью. Мы от самих себя хотим на миг уйти и потому к хмельному склонны зелью.

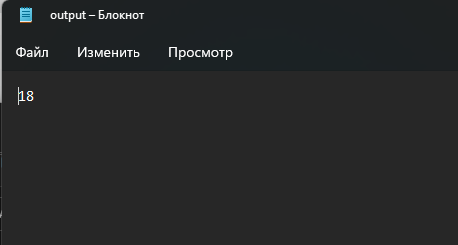
Работа программы:











Варианты пользовательского некорректного ввода:

1) Ввод букв (fhfdggdf)

2) Ввод иных символов (\ . , ? : % ; №)