МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

федеральное государственное автономное образовательное учреждение высшего образования

«САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ   
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

ИНСТИТУТ НЕПРЕРЫВНОГО И ДИСТАНЦИОННОГО ОБРАЗОВАНИЯ

|  |
| --- |
| КАФЕДРА компьютерных технологий и программной инженерии |

ОЦЕНКА

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ |
| «ОБРАБОТКА ТЕКСТОВЫХ ДАННЫХ» |
| по дисциплине: Основы программирования |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | Z9431 |  |  |  | Андреев Д.И. |
|  | номер группы |  | подпись, дата |  | инициалы, фамилия |
| Студенческий билет № | 2019/3781 | |  |  |  |

Санкт-Петербург 2020

Оглавление

[1. Цель работы 3](#_Toc55642159)

[2. Задание 3](#_Toc55642160)

[3. Описание созданных функций 3](#_Toc55642161)

[4. Листинг программы 4](#_Toc55642162)

[5. Пример выполнения программы 6](#_Toc55642163)

[6. Анализ результатов и выводы 6](#_Toc55642165)

1. Цель работы

Целью работы является изучение представления строковых данных и алгоритмов их обработки

1. Задание

Вариант 1

Разработать функцию, которая подсчитывает количество слов в строке.

1. Описание созданных функций

Для решения поставленной в лабораторной работе задачи потребуются следующие функции:

Имя: showWordsNum

Назначение: вычисляет количество слов в строке

Входные данные:

* str – исходная строка

Выходные данные: отсутствуют.

Побочный эффект: отсутствует.

Тестовые данные:

In England the first of April is a very special day. This is the day when people make April’s

Результат выполнения функции:

"В строке: 19 слов"

Прототип: void showWordsNum(const char \*str);

Алгоритм:

Псевдокод:

Объявить переменную count для подсчета количество слов в строке

Для каждого символа в строке

Если данный символ – пробел или символ табуляции

Увеличить счетчик слов на 1

Увеличить счетчик слов на 1 после обхода всех символов

Блок-схема:



1. Листинг программы

#include <iostream>

using namespace std;

// проверка на утечки памяти

#define \_CRTDBG\_MAP\_ALLOC

#include <cstdlib>

#include <crtdbg.h>

#define DBG\_NEW new(\_NORMAL\_BLOCK, \_\_FILE\_\_, \_\_LINE\_\_)

#define new DBG\_NEW

//определения функций

void showHint();

short getCommand();

void showString(const char\* str);

void showWordsNum(const char\* str);

const char\* enterString(const char \*str, size\_t size);

int main()

{

setlocale(0, "Russian");

cout << "Лабораторная работа №6.\nОбработка числовых матриц.\nВариант " << (1 - 1) % 30 + 1 << endl;

short command = -1;

const size\_t size = 1000;

const char\* string = new char[size];

do {

command = getCommand();

switch (command)

{

case 0:

string = enterString(string, size);

break;

case 1:

showString(string);

break;

case 2:

showWordsNum(string);

break;

case 3:

showHint();

break;

case 4:

break;

default:

cout << "Неизвестная команда\n";

break;

}

} while (command != 4);

delete[] string;

// Для обнаружения утечек памяти

\_CrtSetReportMode(\_CRT\_WARN, \_CRTDBG\_MODE\_FILE);

\_CrtSetReportFile(\_CRT\_WARN, \_CRTDBG\_FILE\_STDOUT);

\_CrtSetReportMode(\_CRT\_ERROR, \_CRTDBG\_MODE\_FILE);

\_CrtSetReportFile(\_CRT\_ERROR, \_CRTDBG\_FILE\_STDOUT);

\_CrtSetReportMode(\_CRT\_ASSERT, \_CRTDBG\_MODE\_FILE);

\_CrtSetReportFile(\_CRT\_ASSERT, \_CRTDBG\_FILE\_STDOUT);

\_CrtDumpMemoryLeaks();

return 0;

}

short getCommand()

{

short com;

cout << "Введите команду для продолжения (3 - показать все команды): ";

cin >> com;

while (!cin) {

cin.clear();

cin.ignore(numeric\_limits<streamsize>::max(), '\n');

cout << "Введите корректное значение (4 для выхода): ";

cin >> com;

}

cin.ignore(numeric\_limits<streamsize>::max(), '\n');

return com;

}

void showString(const char\* str)

{

cout << "Текущая строка:\n" << str << '\n';

}

void showWordsNum(const char\* str)

{

unsigned int count = 0;

for (int i = 0; str[i] != '\0'; ++i)

if (str[i] == '\t' || str[i] == ' ')

++count;

++count;

cout << "В строке:\n" << str << "\n" << count << " слов\n";

}

const char\* enterString(const char\* str, size\_t size)

{

delete[] str;

char\* newStr = new char[size];

size\_t i = 0;

cout << "Введите строку: \n";

cin.getline(newStr, size);

return newStr;

}

void showHint()

{

cout << "0 - Ввести строку с клавиатуры\n";

cout << "1 - Вывести строку на экран\n";

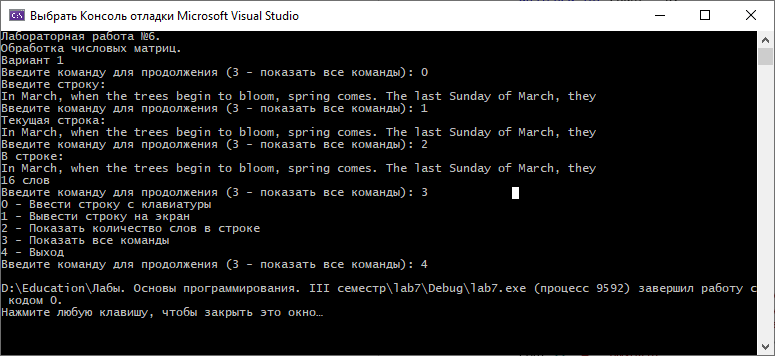
cout << "2 - Показать количество слов в строке\n";

cout << "3 - Показать все команды\n";

cout << "4 - Выход\n";

}

1. Пример выполнения программы



1. Анализ результатов и выводы

К достоинствам программы можно отнести:

Программа работает без ошибок и верно выполняет поставленную задачу

Простота реализации и легко воспринимаемый код программы, а также реализация заданий в виде функций дает возможность повторного использования кода в других проектах

За счет динамического выделения памяти есть возможность выполнения поставленной задачи для строк различной длины

К недостаткам можно отнести:

Линейный поиск, реализованный в данной программе не дает хороших результатов при большой длине строк