ЗВІТ З ЛАБОРАТОРНОЇ РОБОТИ №10

за курсом «Інформатика і програмування»

студента групи ПА-18-3

Дуброва Романа Олеговича

Кафедра комп’ютерних технологій, ДНУ

2018/2019

**1.** *Постановка задачі:*

**Вариант 12**

**1.** Обробка текстової інформації, що міститься у масиві символів

(12) Знайти саме довге слово в реченні та надрукувати його в зверненому вигляді.

**2.** Рядкові функції.

(Написати програму, в якій були б розроблені 2 функції, що імітують дії стандартних рядкових функцій (див. індивідуальне завдання). Вибір функцій здійснюється за допомогою пунктів меню)

Варіант №12 : strpbrk( ) strtol( )

**3.** Обробка текстової інформації, що міститься у файлах.

Завдання:

1. За допомогою текстового редактора створити файл, що містить текст. Довжина рядка тексту не повинна перевищувати 80 символів. Це вхідний файл. 2. Програма повинна 1) інформацію вхідного файлу записати у вихідний файл; 2) реалізувати дії, вказані в індивідуальному завданні і занести відповідну інформацію у вихідний файл.

3. Ім'я вхідного файлу задає користувач в командному рядку або під час виконання програми.

4. Ім'я вихідного файлу: перші символи - не більше трьох символів з імені вхідного файлу, а решти символів \_out.

5. Вихідний файл повинен мати розширення .dat.

6. Програма повинна забезпечувати по запиту користувача: 1) виведення тексту вхідного файлу на екран дисплея; 2) виведення тексту вихідного файлу на екран дисплея.

7. Закінчені послідовності дій оформити у вигляді функцій. Всі необхідні дані для функцій повинні передаватися їм як параметри. Використання глобальних змінних у функціях не допускається.

8. Для вводу і виводу інформації (і для консолі, і для текстових файлів) застосувати засоби форматованого вводу/виводу.

9. Визначення функцій і головну функцію розмістити в двох окремих файлах.

**(№12)** Визначити кількість рядків, що складаються із заданої користувачем кількості слів. Вивести ці рядки.

2. *Опис розв’язку*

Користувачеві пропонується вибрати спосіб введення даних. Залежно від його вибору, дані будуть введені за допомогою файлового потоку або за допомогою консолі. На виході у нас буде два масиви A і B.

Завдання 1: Ми зчитуємо пропозицію і вважаємо довжину кожного слова, порівнюючи з довжиною максимального, яка спочатку дорівнює 0. При цьому ми так само запам'ятовуємо індекс першого елемента слова. Після виходу з циклу ми маємо розмір найбільшого слова і його перший індекс. На основі цієї інформації виводимо слово навпаки.

Завдання 2: Друге завдання у нас складається з 2 двох функцій: strtol і strpbrk. Перша переводить рядок в будь-якій системі числення від 1 до 34 в десяткову систему числення. Користувач вводить базу системи числення в якій він ввів число. Далі, ми вважаємо довжину рядка і вводимо змінну, яка буде відповідати за ступінь розряду. Ми починаємо аналізувати число з кінця, так як там починається розряд 1. Щоб перевести цифру в потрібне значення десяткової системи, ми забираємо певне число, згідно з таблицею ASCII. Далі ми множимо отримане число на ступінь. І множимо ступінь на базу. По завершенні ми повертаємо нашу число. Друга функція послідовно шукає символи з другого рядка в першому рядку. Вона бере кожен символ з другого рядка і шукає його в першій. Якщо символ знайдений, то вона повертає покажчик на цей символ.

Завдання 3: Для початку ми перевіряємо не ввів користувач назву вхідного файлу. Якщо немає, то ми просимо його ввести цю назву. Далі, ми формуємо вихідний файл з перших 3 символів і рядком "\_out.dat" і передаємо імена файлів в функцію menu (). Там ми маємо три функції. Перші дві це висновок вхідного і вихідного файлів. Кожен з файлів виводить одна функція TextDisplay. Третя функція запитує у користувача число, яке буде означати кількість слів у реченні. У вихідний файл функція виводить ті рядки з входного файлу в яких всього 3 слова. Роздільником між словами є багаторазовий пробіл.

3. Вихідний текст програми Програма №1 (завдання 1 і 2)

#include "pch.h"

#include <iostream>

using namespace std;

void firstTask() {

cout << "Ведите макс. кол-во символов, которое будет в вашем предложении: ";

int array\_size;

cin >> array\_size;

char\* array = new char[array\_size];

cin.ignore();

cout << "Введите ваше предложение: ";

cin.getline(array, array\_size);

int i = -1, counter = 0, max\_counter = 0, max\_index = 0;

while (array[i] != '\0') {

i++;

if (array[i] == ' ' || array[i] == '\0') {

if (counter > max\_counter) {

max\_counter = counter;

max\_index = i - counter;

}

counter = 0;

}

else {

counter++;

}

}

cout << "Самое длинное слово из вашего предложения наоборот: ";

for (int i = max\_index + max\_counter - 1; i >= max\_index; i--) {

cout << array[i];

}

cout << endl;

delete[] array;

}

char\* strpbrk(char\* s1, char\* s2) {

int i = 0;

while (s2[i] != '\0') {

int j = 0;

while (s1[j] != '\0') {

if (s2[i] == s1[j]) {

return s1 + j;

}

j++;

}

i++;

}

return NULL;

}

long strtol(char\* str, int base) {

int power = 1, sign = 0;

int i = 0;

if (str[i] == '-') {

sign = 1;

}

long res = 0;

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

for (int j = i - 2; j >= 0 + sign; j--) {

if ((str[j] <= 'Z') && (str[j] >= 'A')) {

res += (str[j] - 55) \* power;

power \*= base;

}

else if ((str[j] <= 'z') && (str[j] >= 'a')) {

res += (str[j] - 87) \* power;

power \*= base;

}

else if ((str[j] <= '9') && (str[j] >= '0')) {

res += (str[j] - 48) \* power;

power \*= base;

}

}

if (sign) res \*= -1;

return res;

}

void SecondFunction() {

cout << "Введите базу системы счисления в которой Вы будете вводить число: ";

int base;

cin >> base;

cout << "Введите кол-во цифр в вашем числе: ";

int size;

cin >> size;

char\* number = new char[size];

cout << "Введите ваше число: ";

cin >> number;

cout << "Ваше число в десятичной системе счичления: " << strtol(number, base) << endl;

}

void FirstFunction() {

cout << "Введите размер первой строки: ";

int n;

cin >> n;

char\* s1 = new char[n + 1];

cout << "Введите первую строку: ";

cin >> s1;

cout << "Введите размер второй строки: ";

int m;

cin >> m;

char\* s2 = new char[m + 1];

cout << "Введите вторую строку: ";

cin >> s2;

cout << "Часть первой строки, которая начинается с первого общего символа из второй строки: ";

char\* temp = strpbrk(s1, s2);

if (temp == NULL) {

cout << "Нет общих символов.\n";

}

else {

cout << temp << endl;

}

delete[] temp;

delete[] s2;

//delete[] s1;

}

int secondTask() {

while (true) {

cout << "1 - Функция strpbrk( ).\n";

cout << "2 - Функция strtol( ).\n";

cout << "0 - Выход\n";

cout << "Выберите пункт: ";

int n;

cin >> n;

switch (n) {

case 1:

system("cls");

FirstFunction();

break;

case 2:

system("cls");

SecondFunction();

break;

case 0:

system("cls");

return 0;

default:

system("cls");

cout << "Такого пункта нет\n";

break;

}

}

}

int main() {

setlocale(LC\_ALL, "Russian");

while (true) {

cout << "1 - Найти самое длинное слово в строке и перевернуть его.\n";

cout << "2 - Реализация 2 стандартных фукнций.\n";

cout << "0 - Выход\n";

cout << "Выберите пункт: ";

int n;

cin >> n;

switch (n) {

case 1:

system("cls");

firstTask();

break;

case 2:

system("cls");

secondTask();

break;

case 0:

return 0;

default:

system("cls");

cout << "Такого пункта меню нет\n";

break;

}

}

return 0;

}

Програма №2 (завдання 3)

#include "pch.h"

#include <iostream>

#include <fstream>

using namespace std;

void TextDisplay(char name[100]) {

ifstream fin(name);

char symbol;

fin >> noskipws;

while (fin >> symbol) {

cout << symbol;

}

cout << endl;

fin.close();

}

int WordsAmount(char array[82]) {

int counter = 0, i = 1;

while (array[i] != '\0') {

if ((array[i] == ' ' && array[i - 1] != ' ') || (array[i] == '\n' && array[i - 1] != ' ')) {

counter++;

}

i++;

}

if (array[i - 1] != ' ' && array[i - 1] != '\n') counter++;

return counter;

}

void IndividualTask(char input\_name[100], char output\_name[100]) {

cout << "Введите нужное вам количество слов: ";

int size;

cin >> size;

char array[82];

int i = 0;

ofstream fon(output\_name);

ifstream fin(input\_name);

fin >> noskipws;

while (fin >> array[i]) {

if (array[i] == '\n') {

array[i + 1] = '\0';

int temp = WordsAmount(array);

if (temp == size) {

fon << array;

}

i = 0;

continue;

}

i++;

}

array[i] = '\0';

int temp = WordsAmount(array);

if (temp == size) {

fon << array;

}

fin.close();

fon.close();

}

int menu(char input\_name[100], char output\_name[100]) {

while (true) {

cout << "Меню\n";

cout << "1 - Вывести содержание входного файла\n";

cout << "2 - Вывести содержание выходного файла\n";

cout << "3 - Найти все предложения с определенным количеством слов\n";

cout << "0 - Выход\n";

cout << "Выберите пункт: ";

int n;

cin >> n;

switch (n) {

case 1:

system("cls");

TextDisplay(input\_name);

break;

case 2:

system("cls");

TextDisplay(output\_name);

break;

case 3:

system("cls");

IndividualTask(input\_name, output\_name);

break;

case 0:

return 0;

default:

cout << "Такого пункта нет!\n";

break;

}

}

}

int main(int args, char\* argv[]) {

setlocale(LC\_ALL, "Russian");

int i = 0;

char input\_name[100];

if (args == 1) {

cout << "Введите имя входного файла: ";

cin >> input\_name;

}

else {

while (argv[1][i] != '\0') {

input\_name[i] = argv[1][i];

i++;

}

input\_name[i] = '\0';

}

char output\_name[100];

for (i = 0; i < 3; i++) {

if (input\_name[i] == '.') {

break;

}

else {

output\_name[i] = input\_name[i];

}

}

char text[9] = "\_out.dot";

for (int j = 0; j <= 9; j++) {

output\_name[i + j] = text[j];

}

menu(input\_name, output\_name);

return 0;

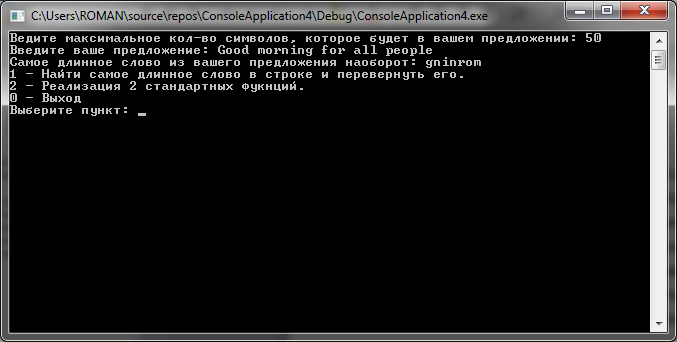
}

*4. Опис інтерфейсу:*

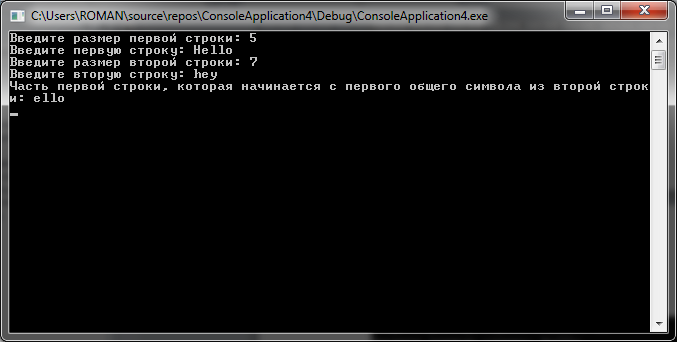
У першій програмі нам пропонується вибрати перше, друге завдання, а так само надається функція виходу з програми. У разі вибору другого завдання, користувач повинен вибрати одну з функції або вийти в попереднє меню. У другій програмі користувач може ввести ім'я вхідного файлу, якщо він не був введений через командний рядок. Далі надається вибір чотирьох пунктів: висновок вхідного файлу, висновок вихідного файлу, індивідуальне завдання і вихід з програми.

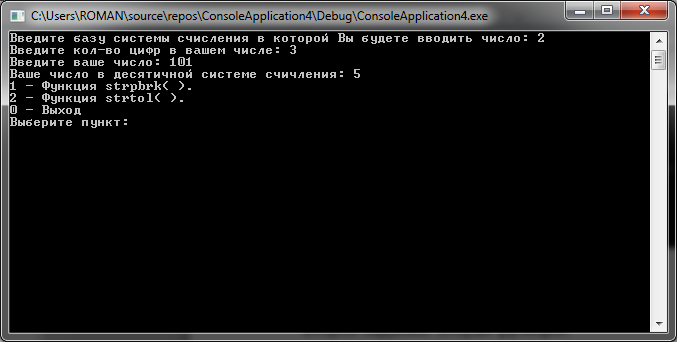
5. *Опис тестових приладів:*

*1-ше завдання*



*2-ше завданння, дві функції*





*3-є завдання, текстовий файл input.txt*

