Белорусский государственный технологический университет

Факультет информационных технологий

Кафедра программной инженерии

 Лабораторная работа 16

По дисциплине «Основы алгоритмизации и программирования»

На тему «Функции пользователя»

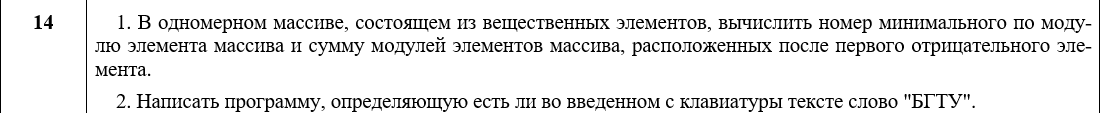
Выполнил:

Студент 1 курса 10 группы

Сегренёв Кирилл Сергеевич

Преподаватель: асс. Андронова М.В.

2023, Минск

**ВАРИАНТ 14**

**Код:**

#include <iostream>

#include <string>

#include <Windows.h>

// 14

using namespace std;

// Прототипы функций

int bstu(string, int);

int minArray(float[], int);

int sumArray(float[], int);

int main() {

setlocale(LC\_ALL, "ru");

SetConsoleCP(1251);

int n;

string text;

int mark;

size\_t g;

cout << "1. Задание 1(Одномерный массив)" << '\n' << "2. Задание 2(Поиск слова БГТУ)" << endl;

cin >> n;

switch (n) {

//======================== 1 ЗАДАНИЕ ==================

case 1:

float\* arr;

int sizeArray;

cout << "Введите размер массива: " << endl;

cin >> sizeArray;

arr = new float[sizeArray]; // Объявляем динамический массив

for (int i = 0; i < sizeArray; i++) {

cout << "Введите " << i + 1 << " элемент последовательности: ";

cin >> arr[i];

}

cout << "min: " << minArray(arr, sizeArray) << endl; // Выводим минимальный элемент последовательности, который считаем в функции minArray

if (sumArray(arr, sizeArray) != 0) // С помощью функции считаем сумму модулей после первого отрицательного элемента, если они есть, вывести данную суму

{

cout << "sum: " << sumArray(arr, sizeArray) << endl;

}

else {

cout << "Нет отрицательных элементов" << endl; // Если нет, вывести данной сообщение

}

cout << "Последовательность: " << endl;

for (int i = 0; i < sizeArray; i++) {

cout << arr[i] << ' ';

}

delete[] arr;

break;

//======================== 2 ЗАДАНИЕ ==================

case 2:

cout << "Введите текст: ";

cin >> text; // Вводим наш текст

g = text.size(); // С помощью функции size считаем длину нашего текста

mark = bstu(text, g); // С помощью функции bstu находим переменную mark(реагирует на то, есть ли БГТУ в тексте, или нет)

if (mark == 1) {

cout << "В данном тексте есть слово БГТУ!" << endl;

}

else {

cout << "В данном тексте нет слов БГТУ :(" << endl;

}

break;

default:

cout << "Введите корректо номер задания!" << endl;

}

return 0;

}

int minArray(float arr[], int size) {

float min = arr[0]; // Присваиваем переменной значение первого элемента последовательности

int indexMin = 0; // Индекс минимального элемента

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

if (min > arr[i]) {

indexMin = i;

min = arr[i];

}

}

return indexMin; // Возвращаем из функции индекс минимального элемента

}

int sumArray(float arr[], int size) {

float sum = 0;

int index = 0;

for (int i = 0; i < size; i++) { // С помощью данного цикла находим индекс первого элемента

if (arr[i] < 0) {

index = i;

break;

}

}

if (index != 0)

{

for (int i = 0; i + index + 1 < size; i++) {

sum += abs(arr[i + index + 1]); // Вычисляем самму модулей элементов, стоящих после отрицательного элемента

}

}

else {

sum = 0;

}

return sum;

}

int bstu(string y, int g){

int mark = 0;

for (int i = 0; i < g; i++) { // Цикл, который по буквам вычисляет, будет ли слово БГТУ

if (y[i] == 'Б') {

if (y[i + 1] == 'Г') {

if (y[i + 2] == 'Т') {

if (y[i + 3] == 'У') {

mark++; // Переменная, реагирующая, что это слово есть

break;

}

}

}

}

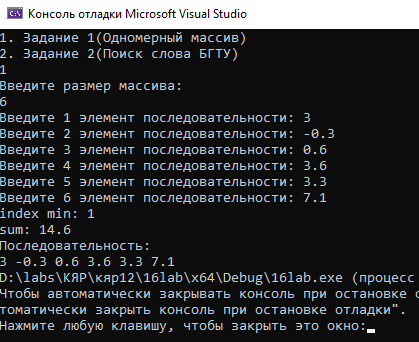
}

return mark;

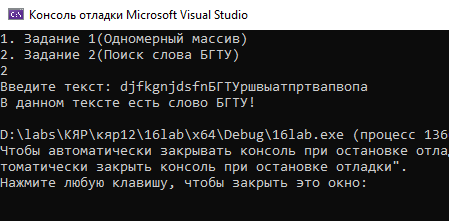
}

**Output**

**Задание 1**

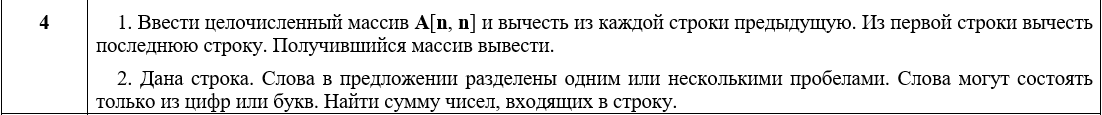
****

**Задание 2**

****

**ДОПОЛНИТЕЛЬНЫЕ ЗАДАНИЯ**

**ВАРИАНТ 4**

****

**Код:**

#include <iostream>

#include <string>

#include <Windows.h>

// 4

using namespace std;

// Прототипы функций

void subtr(int\*\*, int\*\*, int n); // Извините, что использую указатели, но я не знаю, как здесь можно передать через параметр(((((((

int sumNumbers(string, int);

int sum = 0;

int main() {

setlocale(LC\_ALL, "ru");

SetConsoleCP(1251);

int k;

char text[10000];

int g;

cout << "1.Задание 1 (Матрица)" <<'\n' << "2.Задание 2 (Сумма цифр из текста)" << endl;

cin >> k;

cin.ignore(); // Пропускаем символ '\n'

switch (k) {

//======================== 1 ЗАДАНИЕ ==================

case 1: {

cout << "Введите размерность квадратной матрицы: ";

int n;

cin >> n;

int\*\* arr = new int\*[n];

for (int i = 0; i < n; i++)

arr[i] = new int [n];

// Во второй массив записываем ту же матрицу для того, чтобы выполнять действия со старыми элементами матрицы, а не измененными

int\*\* arrOld = new int\* [n];

for (int i = 0; i < n; i++)

arrOld[i] = new int [n];

cout << "Введите элементы матрицы: " << endl;

for (int i = 0; i < n; i++) {

for (int g = 0; g < n; g++) {

cin >> arr[i][g];

arrOld[i][g] = arr[i][g];

}

}

cout << "Исходная матрица: " << endl;

for (int i = 0; i < n; i++) {

for (int g = 0; g < n; g++) {

cout << arr[i][g] << '\t';

}

cout << '\n';

}

subtr(arr, arrOld, n); // Функция subtr вычитает элементы прошлых строк

// Вывод результата вычитания

cout << "После вычитания:" << endl;

for (int i = 0; i < n; i++) {

for (int g = 0; g < n; g++) {

cout << arr[i][g] << '\t';

}

cout << '\n';

}

for (int i = 0; i < n; i++)

{

delete[] arr[i];

}

delete[] arr;

for (int i = 0; i < n; i++)

{

delete[] arrOld[i];

}

delete[] arrOld;

}

break;

//======================== 2 ЗАДАНИЕ ==================

case 2:

cin.getline(text, 10000);

g = strlen(text);

cout << sumNumbers(text, g); // Функция sumNumbers вычисляет сумму цифр, которые встречаются в тексте

break;

default:

cout << "Ошибка(" << "\nВыберите верный вариант!" <<endl;

}

return 0;

}

int sumNumbers(string text, int g) {

for (int i = 0; i < g; i++) {

if (text[i] >= '1' && text[i] <= '9') {

sum += int(text[i]) - '0'; // Преобразование символа в целое число

}

}

return sum;

}

void subtr(int\*\* arr, int\*\* arrOld, int n) {

for (int i = 1; i < n; i++) { // Отнимаем каждую строчку матрицы(кроме первой)

for (int g = 0; g < n; g++) {

arr[i][g] -= arrOld[i - 1][g];

}

}

for (int g = 0; g < n; g++) { // От первой строчки отнимаем последнюю

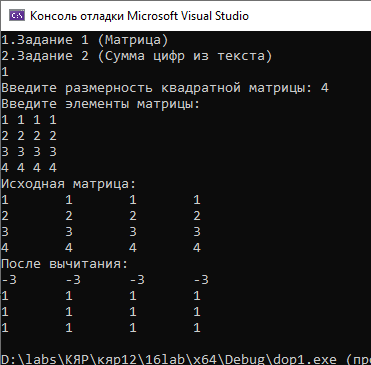
arr[0][g] -= arrOld[n-1][g];

}

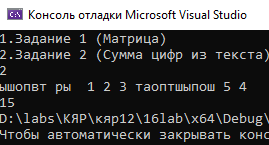
}

**Output:**

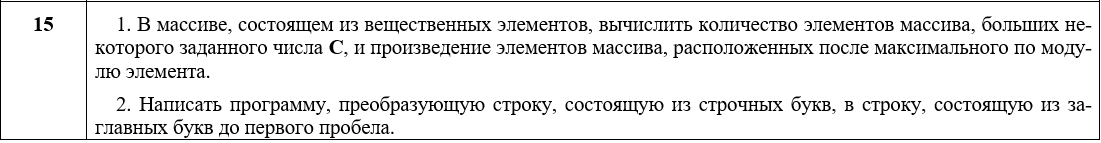
**Задание 1**

****

**Задание 2**

****

**ВАРИАНТ 15**

****

**Код:**

#include <iostream>

#include <string>

#include <Windows.h>

// 15

using namespace std;

// Прототипы функций

int multiplication(float\*, int, int);

int colC(float\*, int, float);

int indexMaxArray(float\*, int);

string upWord(string, int);

int main() {

setlocale(LC\_ALL, "ru");

SetConsoleCP(1251);

int n;

char text[10000];

int index = 0;

int sizeText;

cout << "1. Задание 1(Одномерный массив)" << '\n' << "2. Задание 2(Upper first word)" << endl;

cin >> n;

cin.ignore();

switch (n) {

//======================== 1 ЗАДАНИЕ ==================

case 1:

float\* arr, C;

int sizeArray;

cout << "Введите размер массива" << endl;

cin >> sizeArray;

arr = new float[sizeArray]; // Объявляем динамический массив с указанной длиной

for (int i = 0; i < sizeArray; i++) { // Вводим зачения в массив

cout << "Введите " << i + 1 << " элемент последовательности: ";

cin >> arr[i];

}

cout << "Введите элемент C: " << endl;

cin >> C;

cout << "Количество элементов, больших C: " << colC(arr, sizeArray, C) << endl; // С помощью функции colC найдем, сколько элементов больше переменной C

index = indexMaxArray(arr, sizeArray) + 1; // С помощью indexMaxArray находим индекс элемента, который идет после максимального

cout << "multiplication: " << multiplication(arr, sizeArray, index) << endl; // Со индексом элемента, с которого нужна начинать произведение, заходим в новую функцию, которая и считает наше произведение

for (int i = 0; i < sizeArray; i++) {

cout << arr[i] << ' ';

}

delete[] arr;

break;

//======================== 2 ЗАДАНИЕ ==================

case 2:

cin.getline(text, 10000);

sizeText = strlen(text);

cout << upWord(text, sizeText);

break;

default:

cout << "Ошибка(" << "\nВыберите верный вариант!" << endl;

}

return 0;

}

int colC(float\* arr, int size, float c) {

int col = 0;

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

if (c < arr[i]) {

col++; // При нахождении элемента, большего числа C

}

}

return col; // Вернем это значение

}

int indexMaxArray(float\* arr, int size) {

int index, max = arr[0];

for (int i = 0; i < size; i++) { // Стандартный цикл для нахождения индекса максимального числа(добавился только модуль)

if (abs(arr[i]) > max) {

max = abs(arr[i]);

index = i;

}

}

return index; // Возвращаем индекс максимального числа

}

int multiplication(float\* arr, int size, int y) {

int subtr = 1;

for (; y < size; y++) { // Перемножаем каждое число со значением предыдущих

subtr \*= arr[y];

}

return subtr;

}

string upWord(string text, int g) {

int t = 0, intSymbol;

string wordUp;

for (int i = 0; i < g; i++) {

if (text[i] != ' ') { // Пока не находим первый пробел изменяем слово путем вычитания 32 бит из кода символа

intSymbol = text[i] - 32;

wordUp += char(intSymbol);

t++; // Для того, чтобы понимать, какая длина у слова

}

if (text[i] == ' ') // Когда находим пробел, выходим из цикла

break;

}

for (t; t < g; t++) {

wordUp += text[t]; // Записываем словов в новую переменную

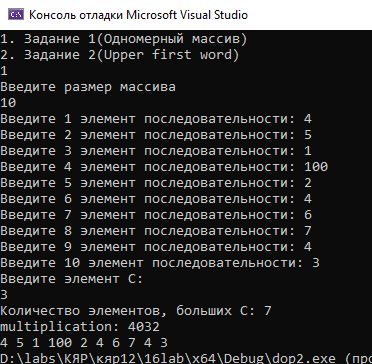
}

return wordUp;

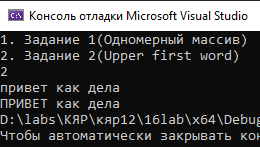
}

**Output:**

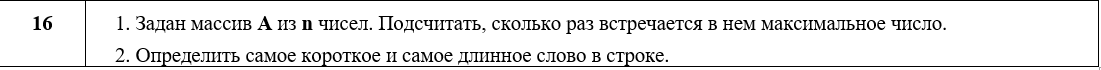
**Задание 1**

****

**Задание 2**

****

**ВАРИАНТ 16**

****

**Код:**

#include <iostream>

#include <string>

#include <Windows.h>

#include <cstring>

//16

using namespace std;

// Прототипы функций

string findShortWord(string, int);

string findLongWord(string, int);

int max\_Number(int\*, int);

int col\_Max\_Numbers(int, int\*, int);

int main() {

setlocale(LC\_ALL, "ru");

SetConsoleCP(1251);

int n;

char text[200];

int indexMin = 0;

int indexMax = 0;

int g;

int maxNumber, colMaxNumbers;

cout << "1. Задание 1(Максимальное число)" << '\n' << "2. Задание 2(Короткое и длиное слово)" << endl;

cin >> n;

cin.ignore(); // Игнорируем \n

switch (n) {

//======================== 1 ЗАДАНИЕ ==================

case 1:

int\* arr;

int sizeArray;

cout << "Введите размер массива" << endl;

cin >> sizeArray;

arr = new int[sizeArray]; // Объявляем динамический массив

for (int i = 0; i < sizeArray; i++) {

cout << "Введите " << i + 1 << " элемент последовательности: ";

cin >> arr[i];

}

maxNumber = max\_Number(arr, sizeArray); // С помощью функции вычисляем максимальное число

colMaxNumbers = col\_Max\_Numbers(maxNumber, arr, sizeArray); // В данной функции с помощью значения maxNumber находим количество максимальных элементов

cout << "Столько максимальный элемент повторяется в последовательности: " << colMaxNumbers << endl;

delete[] arr;

break;

//======================== 1 ЗАДАНИЕ ==================

case 2:

cin.getline(text, 200);

g = strlen(text);

// С помощью данных функций находим короткое и длинное слово

cout << "Самое короткое слово: " << findShortWord(text, g) << endl;

cout << "Самое длинное слово: " << findLongWord(text, g) << endl;

break;

default:

cout << "Ошибка(" << "\nВыберите верный вариант!" << endl;

}

return 0;

}

int max\_Number(int\* arr, int size) {

int max = arr[0];

for (int i = 0; i < size; i++) { // Нахождение максимального элемента

if (max < arr[i]) {

max = arr[i];

}

}

return max;

}

int col\_Max\_Numbers(int max, int\* arr, int size) {

int col = 0;

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

if (max == arr[i]) {

col++; // Если в массиве встречается максимальный элемент, увеличивать количество переменной

}

}

return col;

}

string findShortWord(string y, int g) {

int lengthWord = 0;

int minWord = g + 2; // Присваиваем значение длины текста, т.к дальше некуда(чтобы находить мин слово)

int id = 0;

string shortWord;

for (int i = 0; i < g; i++) {

if (y[i] != ' ' && y[i] != '\0') {

lengthWord++; // Вычисляем длину слова

}

else {

if (minWord > lengthWord) {

id = i; // Присвоение переменной индекс с конца мин слова

minWord = lengthWord;

}

lengthWord = 0;

}

}

for (int i = id - minWord; i <= id; i++) { // Записываем слово( не с конца, а с начала)

shortWord += y[i];

}

return shortWord;

}

//Алгоритм нахождения длиного слова почти такой же, но все такие есть маленькие изменения

string findLongWord(string y, int g) {

int lengthWord = 0;

int maxWord = 0; // Ниже некуда )

int id = 0;

string longWord;

for (int i = 0; i < g; i++) {

if (y[i] != ' ' && y[i] != '\0') {

lengthWord++;

}

else {

if (maxWord < lengthWord) { // Поставить меньше

id = i;

maxWord = lengthWord;

}

lengthWord = 0;

}

}

for (int i = id - maxWord; i <= id; i++) {

longWord += y[i];

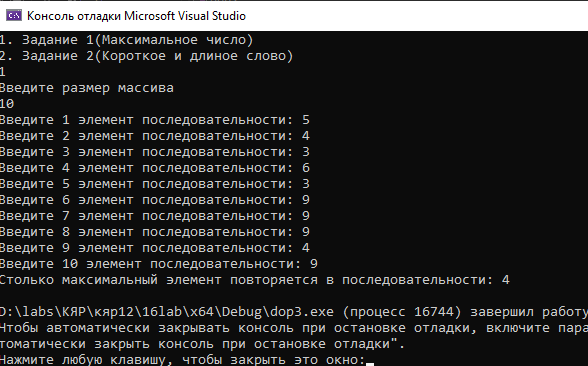
}

return longWord;

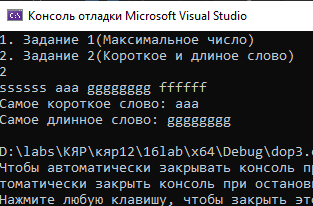
}

**Output:**

**Задание 1**

****

**Задание 2**

****