ГУАП

КАФЕДРА № 43

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

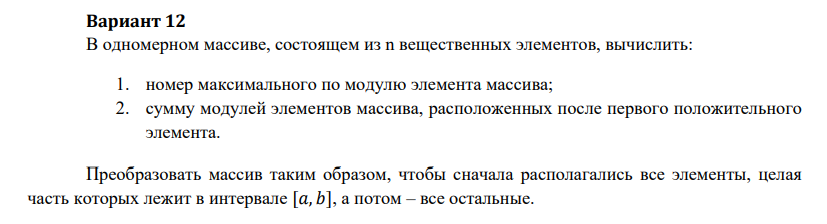
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ассистент |  |  |  | М. А. Мурашова |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ |
| Обработка числовых последовательностей |
| по курсу: ОСНОВЫ ПРОГРАММИРОВАНИЯ |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 4135К |  |  |  | Столяров Н.С. |
|  |  |  | подпись, дата |  | инициалы, фамилия |

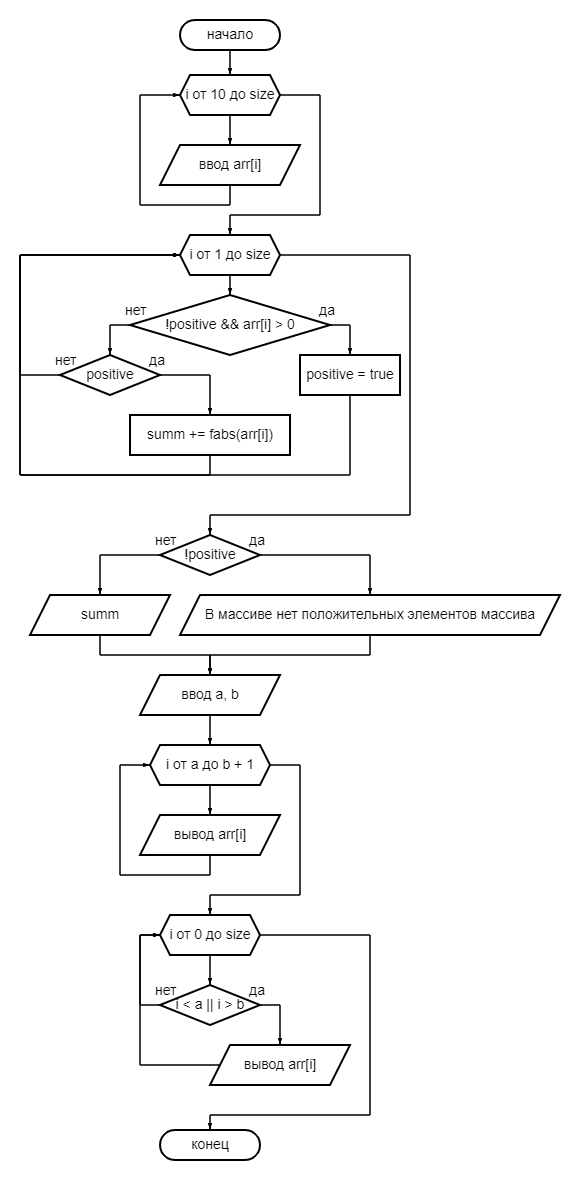
Санкт-Петербург 2022

**1.Цель работы:** Целью работы является изучение структуры данных одномерный массив.  
**2.Задачи работы:** Задания на лабораторную работу приводятся в каждом варианте. При написании программ можно использовать как динамические, так и нединамические массивы. Размерность последних задаѐтся именованной константой.  
  
**3.Описание функций:  
1)Имя: main**   
**Назначение:** Вычисление номера максимального по модулю элемента, суммы модулей элементов массива и преобразование массива.  
**Входные данные:** нет.   
**Выходные данные:** нет.  
**Побочный эффект:** отсутствует.   
**Тестовые данные:**

|  |  |  |  |
| --- | --- | --- | --- |
| Массив | A | B | Ответ |
| 1 2 3 4 5 6 7 8 9 0 | 2 | 5 | 1 2 3 4 5 6 7 8 9 0 3 4 5 6 1 2 7 8 9 0 |

**Прототип:** int main()  
**Псевдокод:**  
Ввод массива arr из 10 элементов  
Вывод массива  
Определение максимального по модулю элемента массива  
Определение суммы элементов после 1 положительного элемента  
Ввод диапазона  
Вывод элементов массива в этом диапазоне  
Вывод остального массива

**Блок-схема:**



**4. Имя: read\_value**  
**Назначение:** проверка переменной a на корректность ввода  
**Входные данные:** Отсутствуют.   
**Выходные данные:** x(double).  
**Побочный эффект:** отсутствует.   
**Прототип:** double read\_value()

|  |  |
| --- | --- |
| **Псевдокод:** | **Блок-схема:** |
| Бесконечный цикл  ввод строки  проверка на пробелы и удаление их  проверка на запятые и замена их на точки  проверка на минус  проверка на лишние символы  если не было проблем, то останавливаем бесконечный цикл­  конец бесконечного цикла  перевод строки в число  вывод числа |  |

**5.ЛИСТИНГ КОДА:**/\*

Задания на лабораторную работу приводятся в каждом варианте. При написании программ

можно использовать как динамические, так и нединамические массивы. Размерность последних

задаѐтся именованной константой.

Вариант 12

В одномерном массиве, состоящем из n вещественных элементов, вычислить:

1. номер максимального по модулю элемента массива;

2. сумму модулей элементов массива, расположенных после первого положительного

элемента.

Преобразовать массив таким образом, чтобы сначала располагались все элементы, целая

часть которых лежит в интервале 𝑎, 𝑏 , а потом – все остальные.

\*/

#include <iostream>

using namespace std;

#include "libs/lib.h"

#include <cmath>

// проверка ввода

#include "libs/simple\_char.h"

#include "libs/input\_validation.h"

int main() {

// смена кодировки

system("chcp 65001");

draw\_line(20);

// ввод элементов массива

int size = read\_value("Введите размер массива: ", false, false, false);

double \*arr = (double\*) malloc(size \* sizeof(double));

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

cout << "array [" << i << "] = ";

arr[i] = read\_value("", true, true, false);

}

draw\_line(20);

// выводим массив

draw\_float\_array(arr, size);

draw\_line(20);

// Номер максимального по модулю элемента массива

int max\_id = 0;

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

if (fabs(arr[i]) > fabs(arr[max\_id]))

max\_id = i;

}

cout << "Номер максимального по модулю элемента массива: " << max\_id << endl;

// Сумма модулей после позитивного элемента

bool positive = false;

float summ = 0;

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

if (!positive && arr[i] > 0) {

positive = true;

} else if (positive){

summ += fabs(arr[i]);

}

}

if (!positive)

cout << "В массиве нет положительных элементов массива\n";

else

cout << "Сумма элементов массива после 1 положительного элемента: " << summ << endl;

draw\_line(20);

// ввод диапазона

int a, b, k;

cout << "Введите диапазон [a, b]\n";

while (true) {

a = read\_value("A = ", false, false, false);

b = read\_value("B = ", false, false, false);

// меняем местами числа, если пользователь их перепутал

if (a > b) {

k = a;

a = b;

b = k;

}

// если числа одинаковые, то вводим заново

if (a == b) {

cout << "A и B не могут быть одинаковыми." << endl;

draw\_line(20);

}else if (a >= size || b >= size) {

cout << "A и B не могут быть больше массива." << endl;

draw\_line(20);

}

else {

// если всё ок выходим из цикла

break;

}

}

// пересобираем массив

double \*new\_arr = (double\*) malloc(size \* sizeof(double));

int j = 0;

for (int i = a; i < b + 1; i++){

new\_arr[i] = arr[j++];

}

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

if (i < a || i > b)

new\_arr[i] = arr[j++];

}

// выводим новый массив

draw\_line(20);

draw\_float\_array(new\_arr, size);

draw\_line(20);

// удаление массивов

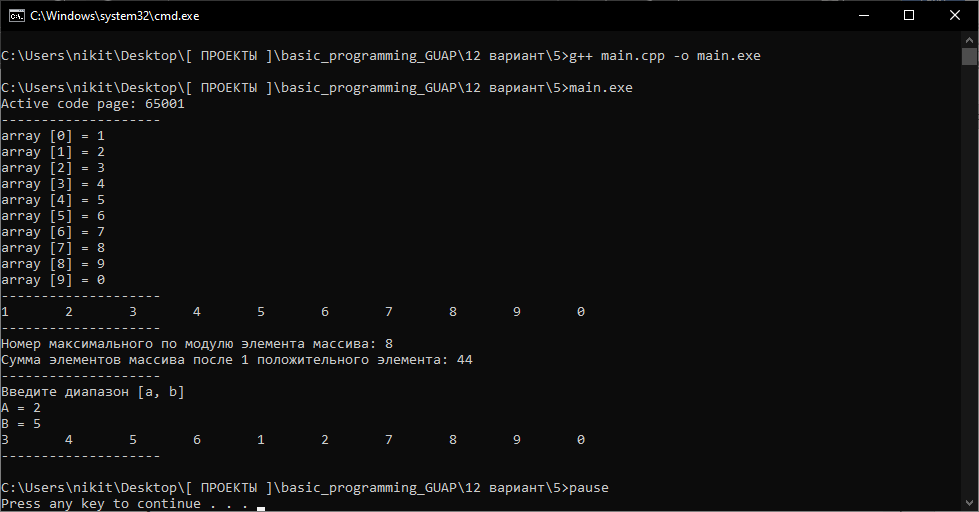
delete(arr);

delete(new\_arr);

return 0;

}

**10.Пример выполнения программы:**



Видно, что результаты расчётов совпадают с тестовыми данными.

**11.Анализ результатов и выводы:**

В ходе этой лабораторной работы мы научились работаь с одномерными массивами.

Из недостатков можно отметить то что многие части кода не разбиты на разные функции.

Из плюсов можно отметить то что оно работает.