

КАФЕДРА №

ОТЧЕТ  
ЗАЩИЩЕН С ОЦЕНКОЙ  
ПРЕПОДАВАТЕЛЬ

\_\_\_\_\_  
должность, уч. степень, звание

\_\_\_\_\_  
подпись, дата

\_\_\_\_\_  
инициалы, фамилия

## ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ

### **Обработка числовых последовательностей**

по курсу: ОСНОВЫ ПРОГРАММИРОВАНИЯ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ ГР. №

\_\_\_\_\_  
подпись, дата

\_\_\_\_\_  
инициалы, фамилия

Санкт-Петербург 2022

**1.Цель работы:** Целью работы является изучение структуры данных одномерный массив.

**2.Задание:** Задания на лабораторную работу приводятся в каждом варианте. При написании программ можно использовать как динамические, так и нединамические массивы. Размерность последних задаётся именованной константой.

Согласно варианту 17:

### **Вариант 17**

В одномерном массиве, состоящем из  $n$  целых элементов, вычислить:

1. количество положительных элементов массива;
2. сумму элементов массива, расположенных после последнего элемента, равного 0.

Преобразовать массив таким образом, чтобы сначала располагались все элементы, целая часть которых не превышает 1, а потом – все остальные.

### **3.Описание созданных функций:**

**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	1234567890
A	9
B	0
Ответ	1023456789

**Прототип:** int main()

**Псевдокод:**

Ввод массива агт из 10 элементов

Вывод массива

получение и вывод количества в массиве положительных элементов

получение и вывод суммы элементов после 0

преобразование массива

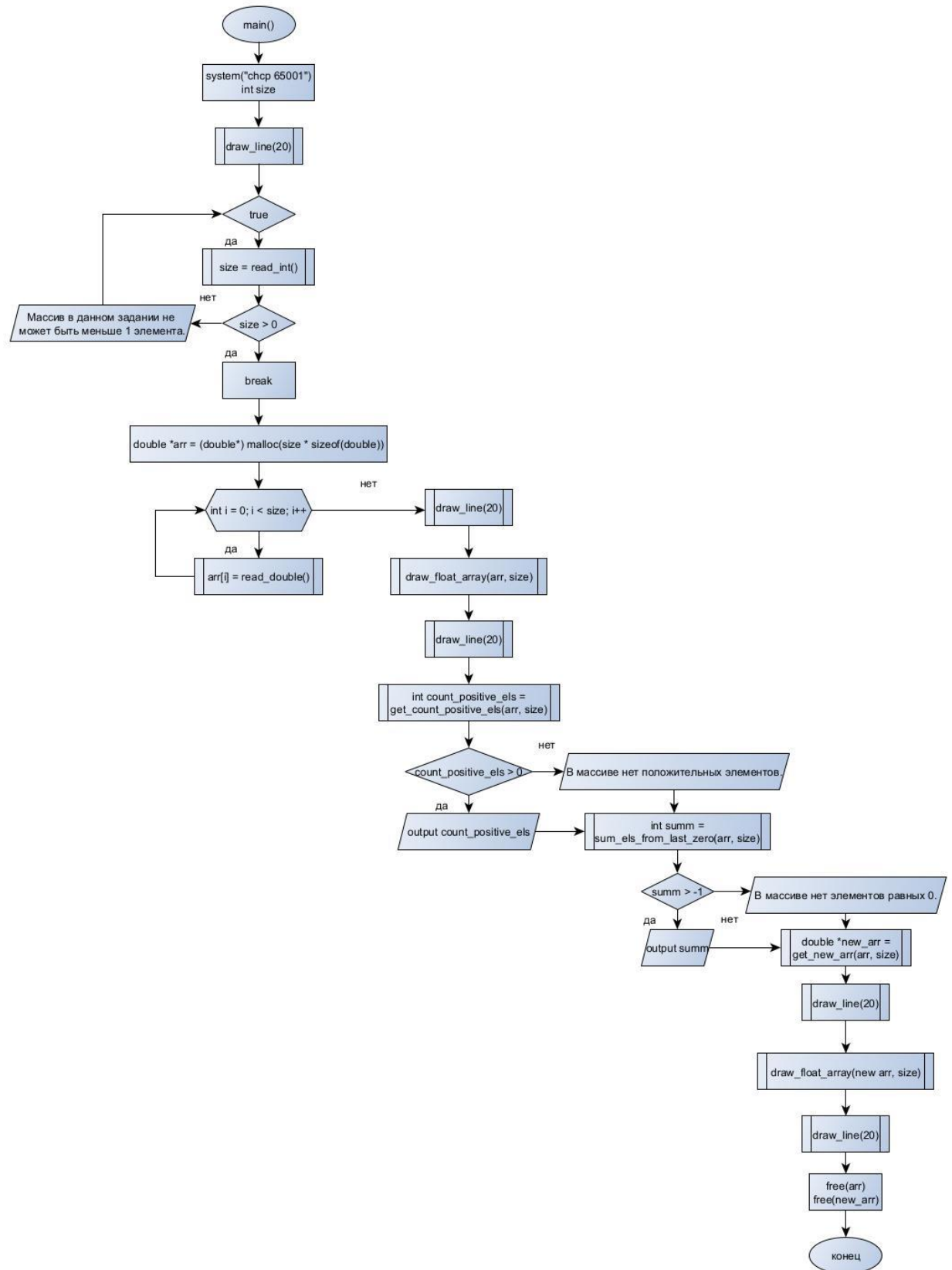
создаётся новый массив

целая часть которых не превышает 1

остальные

возвращение массива

**Блок-схема:**



**Имя:** `get_count_positive_els`

**Назначение:** функция для нахождения в массиве количества положительных элементов

**Входные данные:** `count_positive_els = 0`

**Выходные данные:** `count_positive_els`

**Побочный эффект:** отсутствует.

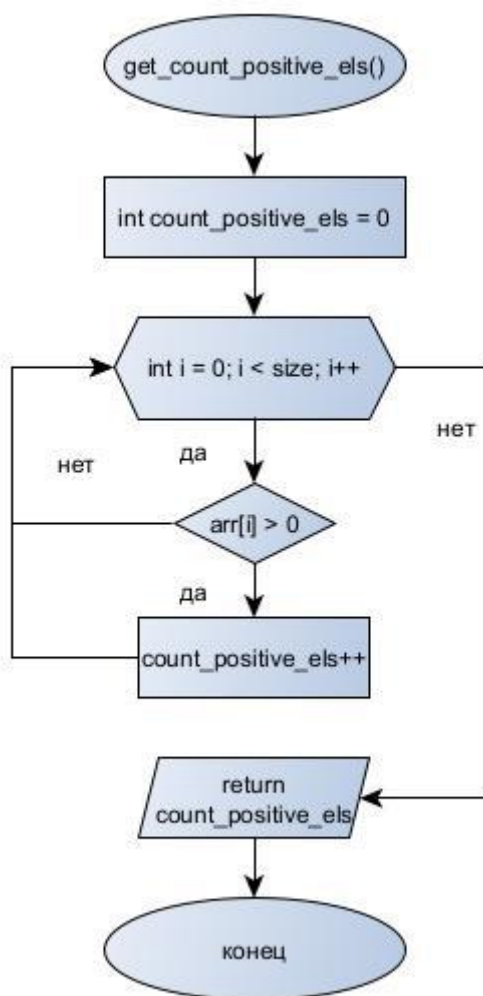
**Прототип:** `int get_count_positive_els()`

**Псевдокод:**

Вводится счётчик приравненный к нулю

Если число больше 0, то к счётчику прибавляется 1

**Блок-схема:**



**Имя:** `sum_els_from_last_zero`

**Назначение:** функция для получения суммы элементов после 0

**Входные данные:**

**Выходные данные:** `summ`

**Побочный эффект:** отсутствует.

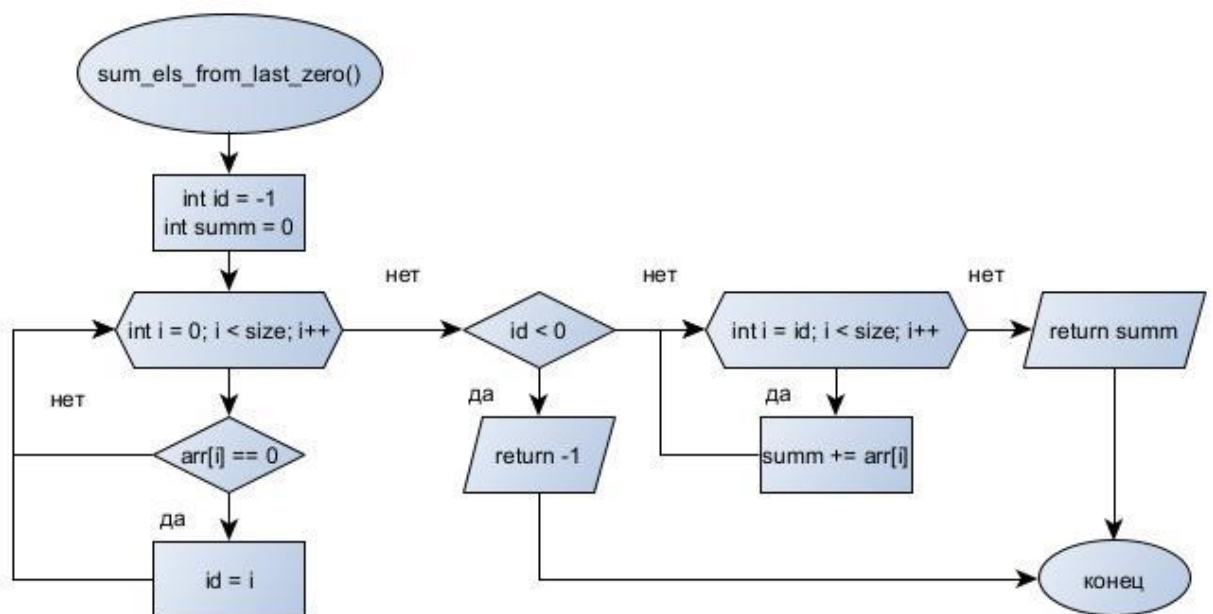
**Прототип:** `int sum_els_from_last_zero`

**Псевдокод:**

Находится 0

Находятся элементы после 0 и складываются

**Блок-схема:**



**Имя:** `get_new_arr`

**Назначение:** преобразование массива

**Входные данные:**

**Выходные данные:**

**Побочный эффект:** отсутствует.

**Прототип:** `double* get_new_arr()`

**Псевдокод:**

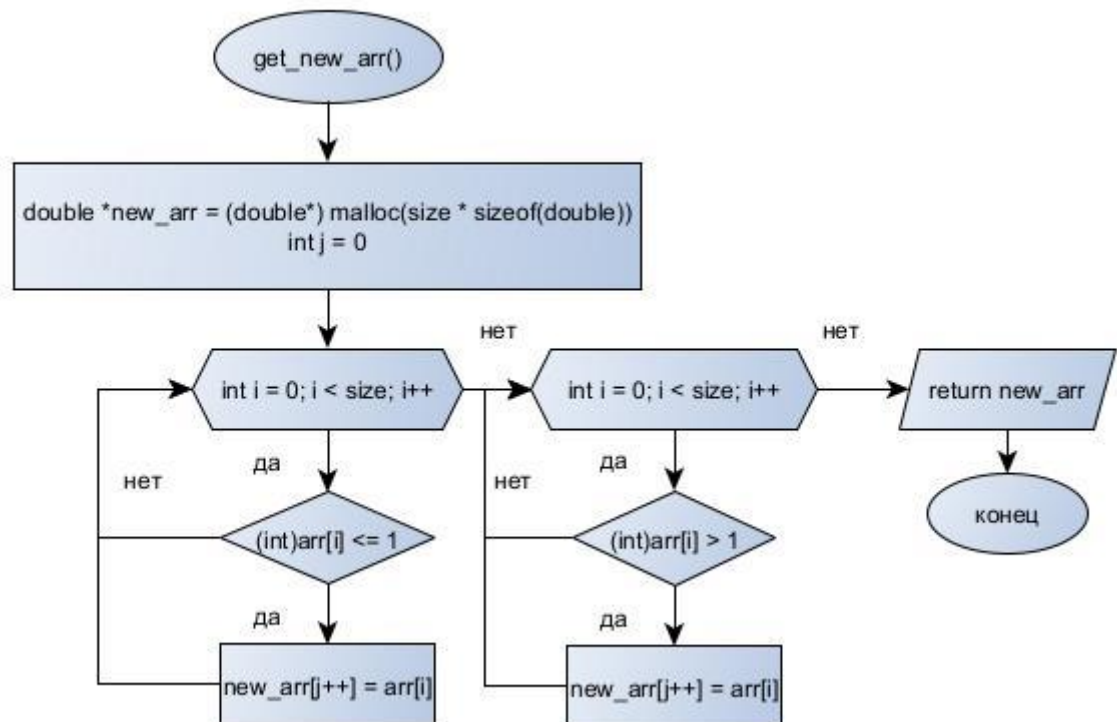
Создаётся новый массив

Находятся числа с целой частью меньше 1 и записываются

Записываются остальные числа

Возвращается массив

**Блок-схема:**



#### 4. ЛИСТИНГ КОДА:

```
#define _CRT_SECURE_NO_WARNINGS
#include <iostream>
using namespace std;

#include "lib.h"
#include <cmath>

// функции для лабы (get_count_positive_els, sum_els_from_last_zero, get_new_arr)
#include "functions.h"

int main() {
    // смена кодировки
```

```

system("chcp 65001");
setlocale(LC_ALL, "Russian");
draw_line(20);

// ввод размера массива с проверкой
int size;
while (true) {
    cout << "Введите размер массива: ";
    size = read_int();

    if (size > 0) {
        break;
    }
    else {
        cout << "Массив в данном задании не может быть меньше 1 элемента." << endl;
    }
}

// создаём массив
double* arr = (double*)malloc(size * sizeof(double));

// вводим массив
for (int i = 0; i < size; i++) {
    cout << "array [" << i << "] = ";
    arr[i] = read_double();
}

draw_line(20);
draw_float_array(arr, size); // выводим массив
draw_line(20);

// получение и вывод количества в массиве положительных элементов
int count_positive_els = get_count_positive_els(arr, size);
if (count_positive_els > 0)
    cout << "Положительных элементов в массиве: " << count_positive_els << endl;
else
    cout << "В массиве нет положительных элементов." << endl;

// получение и вывод суммы элементов после 0
int summ = sum_els_from_last_zero(arr, size);
if (summ > -1)
    cout << "Сумма элементов массива, расположенных после последнего элемента, равного 0: " << summ << endl;
else
    cout << "В массиве нет элементов равных 0." << endl;

double* new_arr = get_new_arr(arr, size);

draw_line(20);
draw_float_array(new_arr, size); // выводим новый массив
draw_line(20);

// удаление массивов
free(arr);
free(new_arr);

return 0;
}

```

## 5. Пример выполнения программы:

```
Консоль отладки Microsoft Visual Studio
Active code page: 65001
-----
Введите размер массива: 10
array [0] = 1
array [1] = 2
array [2] = 3
array [3] = 4
array [4] = 5
array [5] = 6
array [6] = 7
array [7] = 8
array [8] = 9
array [9] = 0
-----
1      2      3      4      5      6      7      8      9      0
-----
Положительных элементов в массиве: 9
Сумма элементов массива, расположенных после последнего элемента, равного 0: 0
-----
1      0      2      3      4      5      6      7      8      9
-----
I:\5\ConsoleApplication1\Debug\ConsoleApplication1.exe (процесс 9272) завершил работу с кодом 0.
Чтобы автоматически закрывать консоль при остановке отладки, включите параметр "Сервис" ->"Параметры" ->"Отладка" -> "Автоматически закрыть консоль при остановке отладки".
Нажмите любую клавишу, чтобы закрыть это окно...
```

## 6. Анализ результатов и выводы:

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

К достоинствам программы можно отнести:

- Производится проверка входных данных.
- Программа корректно работает.

Из недостатков можно отметить:

- Программа не оптимизирована.
- Можно было бы использовать больше функций.