МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное автономное образовательное учреждение высшего образования

«САНКТ-ПЕТЕРБУГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
АЭРОКОСМИЧЕСКОГО ПРИБОРОСТРОЕНИЯ»

КАФЕДРА № 43

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

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

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ № 5 |
| «Обработка числовых последовательностей» |
| по курсу: Основы программирования |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 4134К |  |  |  | Опарин С.Н. |
|  |  |  | подпись, дата |  | инициалы, фамилия |

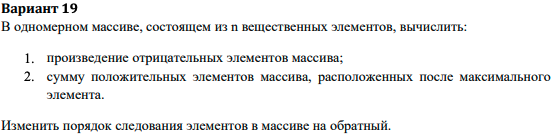
Санкт-Петербург 2022

1. **Цель работы:**

Целью работы является изучение структуры данных одномерный массив

1. **Задание**

Задания на лабораторную работу приводятся в каждом варианте. При написании программ можно использовать как динамические, так и нединамические массивы. Размерность последних задаѐтся именованной константой. Примечание: массивы, созданные с помощью классов (таких, как например, array или vector из библиотеки STL) использовать запрещается.



1. **Описание созданных функций**

Для реализации задания нам потребуется следующие функции:

**Имя:** check

**Назначение:** проверка вводимых данных на корректность

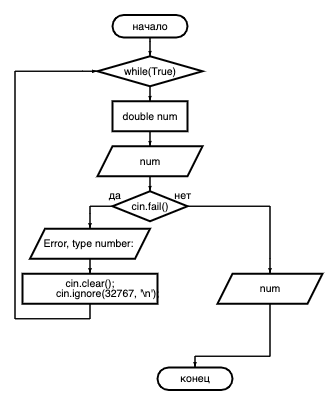
**Входные данные:** ввод с клавиатуры

**Выходные данные:** проверенный ввод с клавиатуры

**Побочный эффект:** отсутствует

**Тестовые данные:**

|  |  |
| --- | --- |
| **num** | **Результат** |
| **Числа** | Число |
| **Символы** | Error, type number: |

****

**Имя:** mas\_fill

**Назначение:** формирование массива

**Входные данные:** размер массива

**Выходные данные:** заполненный массив

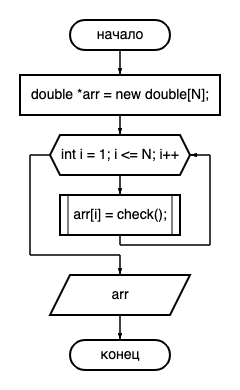
**Побочный эффект:** отсутствует

**Псевдокод:** Инициализируем массив

От 1 до N  
 Формируем значение с помощью функции  
 Верни массив

**Прототип:** double \* mas\_fill(int N)

**Блок-схема:**

****

**Имя:** Arr\_max

**Назначение:** нахождение максимального значения в массиве

**Входные данные:** размер массива, массив

**Выходные данные:** максимальное значение в массиве

**Побочный эффект:** отсутствует

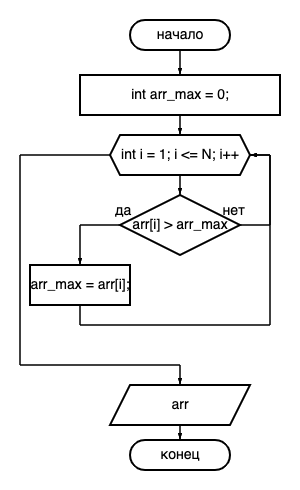
**Псевдокод:** Инициализируем переменные

От 1 до N  
 Если текущее значение больше максимального, то

Переприсвой значение  
 Верни максимальное значение

**Прототип:** double Arr\_max(double arr[], int N)

**Блок-схема:**

****

**Имя:** task\_1

**Назначение:** формирование произведения отрицательных элементов массива

**Входные данные:** размер массива, массив

**Выходные данные:** произведение отрицательных элементов массива

**Побочный эффект:** отсутствует

**Псевдокод:** Инициализируем переменные

От 1 до N  
 Если текущее значение меньше нуля, то

Умножь переменную с произведением на этот элемент

Подними флаг о наличии отрицательных элементов

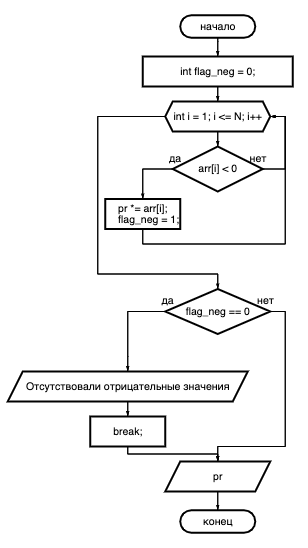
Если флаг не поднят, то

Выведи сообщение о том, что отрицательные значения отсутствовали

Выйди  
 Верни максимальное значение

**Прототип:** double task\_1(double arr[], int N)

**Блок-схема:**

****

**Имя:** task\_2

**Назначение:** формирование суммы положительных элементов после первого вхождения максимального значения массива

**Входные данные:** размер массива, массив, максимальное значение массива

**Выходные данные:** сумма положительных элементов после первого вхождения максимального значения массива

**Побочный эффект:** отсутствует

**Псевдокод:** Инициализируем переменные

От 1 до N  
 Если текущее значение равняется максимальному значению и флаг первого вхождения максимального значения не поднят, то

Подними два флага

Продолжи

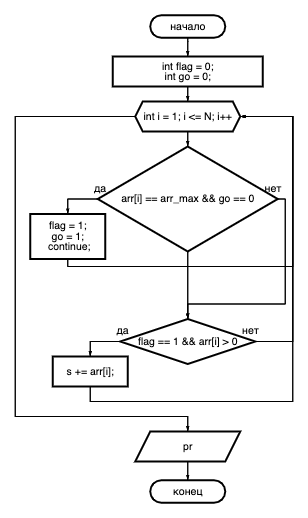
Если флаг поднят и значение массива положительное, то

Прибавь значение массива к переменной с суммой

Верни сумму

**Прототип:** double task\_2(int N,double arr[], int arr\_max)

**Блок-схема:**

****

**Имя:** task\_3

**Назначение:** вывод массива в обратном порядке

**Входные данные:** массив, размер массива

**Выходные данные:** массив в обратном порядке

**Побочный эффект:** отсутствует

**Псевдокод:** Выведи сообщение

От N до 1  
 Выведи элемент

**Прототип:** double task\_2(double arr[], int N)

**Блок-схема:**

****

**Имя:** main

**Назначение:** вызов всех функций и вывод ответов на задания по варианту

**Входные данные:** размер массива

**Выходные данные:** массив в обратном порядке

**Побочный эффект:** отсутствует

**Псевдокод:** Выведи сообщение

Инициализируем переменные

Ввод размера массива

Вывод сообщения

Вызов функций

Вывод ответов

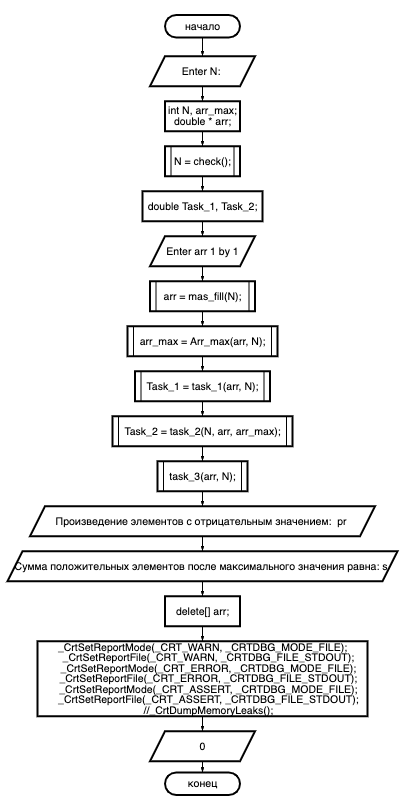
Очищение памяти

**Прототип:** double main()

**Тестовые данные:**

|  |  |
| --- | --- |
| **Входные данные** | **Выходные данные** |
| **3 -5 8 3** | **Массив в обратном порядке:**  **3**  **8**  **-5**  **Произведение элементов с отрицательным значением: -5**  **Сумма положительных элементов после максимального значения равна: 3** |
| **3 7 4 8** | **Отсутствовали отрицательные значения**  **Массив в обратном порядке:**  **8**  **4**  **7**  **Произведение элементов с отрицательным значением: 1**  **Сумма положительных элементов после максимального значения равна: 0** |

**Блок-схема:**

****

1. **Текст программы**

#include <iostream>

#include <string>

#include <stdlib.h>

#define \_CRTDBG\_MAP\_ALLOC

#include <crtdbg.h>

#ifdef \_DEBUG

#ifndef DBG\_NEW

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

#define newDBG\_NEW

#endif

#endif

using namespace std;

int pr = 1, s = 0;

double check() {

while (true) {

double num;

cin >> num;

if (cin.fail()) {

cout << "Error, type number: ";

cin.clear();

cin.ignore(32767, '\n');

}

else {

return num;

}

}

}

double \* mas\_fill(int N) {

double \*arr = new double[N];

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

arr[i] = check();

}

return arr;

}

double Arr\_max(double arr[], int N) {

int arr\_max = 0;

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

if (arr[i] > arr\_max) {

arr\_max = arr[i];

}

}

return arr\_max;

}

double task\_1(double arr[], int N) {

int flag\_neg = 0;

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

if (arr[i] < 0) {

pr \*= arr[i];

flag\_neg = 1;

}

if (flag\_neg == 0) {

cout << "Отсутствовали отрицательные значения" << endl;

break;

}

}

return pr;

}

double task\_2(int N, double arr[], int arr\_max) {

int flag = 0;

int go = 0;

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

if (arr[i] == arr\_max && go == 0) {

flag = 1;

go = 1;

continue;

}

if (flag == 1 && arr[i] > 0) {

s += arr[i];

}

}

return s;

}

void task\_3(double arr[], int N) {

cout << "Массив в обратном порядке: " << endl;

for (int i = N; i >= 1; i--) {

cout << arr[i] << endl;

}

}

int main() {

cout << "Enter N: ";

int N, arr\_max;

double \* arr;

N = check();

double Task\_1, Task\_2;

cout << "Enter arr 1 by 1" << endl;

arr = mas\_fill(N);

arr\_max = Arr\_max(arr, N);

Task\_1 = task\_1(arr, N);

Task\_2 = task\_2(N, arr, arr\_max);

task\_3(arr, N);

cout << "Произведение элементов с отрицательным значением: " << pr << endl;

cout << "Сумма положительных элементов после максимального значения равна: " << s << endl;

delete[] arr;

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;

}

1. **Пример выполнения программы**

Ниже показан пример выполнения программы

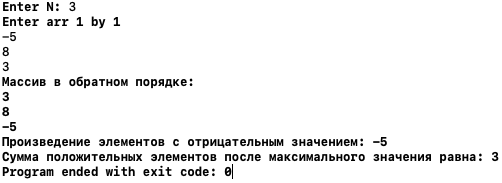


Рис. 1 - Пример выполнения программы

Видно, что результаты расчетов совпадают с тестовыми данными.

1. **Анализ результатов и выводы**

В результате выполнения лабораторной работы были изучены структуры данных одномерного массива

К достоинствам программы можно отнести:

* Решение каждого выражения реализовано в виде отдельной функции
* Есть проверка на корректность данных

Из недостатков можно отметить:

* Недостаточно оптимизирован код