

ГУАП

КАФЕДРА № 42

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

доцент

должность, уч. степень, звание

подпись, дата

В. А. Кузнецов

инициалы, фамилия

ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ № 1.3

ОДНОМЕРНЫЕ МАССИВЫ

по курсу:

ОСНОВЫ ПРОГРАММИРОВАНИЯ

РАБОТУ ВЫПОЛНИЛ

СТУДЕНТ гр. №

4326

подпись, дата

Г. С. Томчук

инициалы, фамилия

Санкт-Петербург 2024

СОДЕРЖАНИЕ

1	Постановка задачи.....	3
2	Схема алгоритма решения.....	4
3	Полное описание реализованной функции.....	5
3.1	exclude_max.....	5
3.2	read_array.....	5
3.3	write_array.....	6
3.4	main.....	6
4	Листинг программы.....	7
5	Несколько тестов работы программы.....	8

1 Постановка задачи

Задача: реализовать программную функцию на языке C/C++, выполняющую поставленную задачу. Вариант задания, пример входных и выходных данных представлен в таблице 1. Глобальные параметры использовать запрещено; допустимо использование дополнительных функций. Использовать только динамические массивы, ввод и вывод данных должен быть осуществлен в текстовый файл. Во входном и выходном файле необходимо указать размерность массива (массивов), размерность выходного массива определяется реализуемой функцией и является возвращаемым значением.

Таблица 1 – Вариант

N	Текст задания	Вход	Выход
7	Реализовать функцию, формирующую выходной массив В на основе входного массива А, исключив из него все элементы с максимальным значением в массиве А.	10 4,2,4,1,3, 0,1,3,4,1	7 2,1,3,0,1, 3,1

2 Схема алгоритма решения

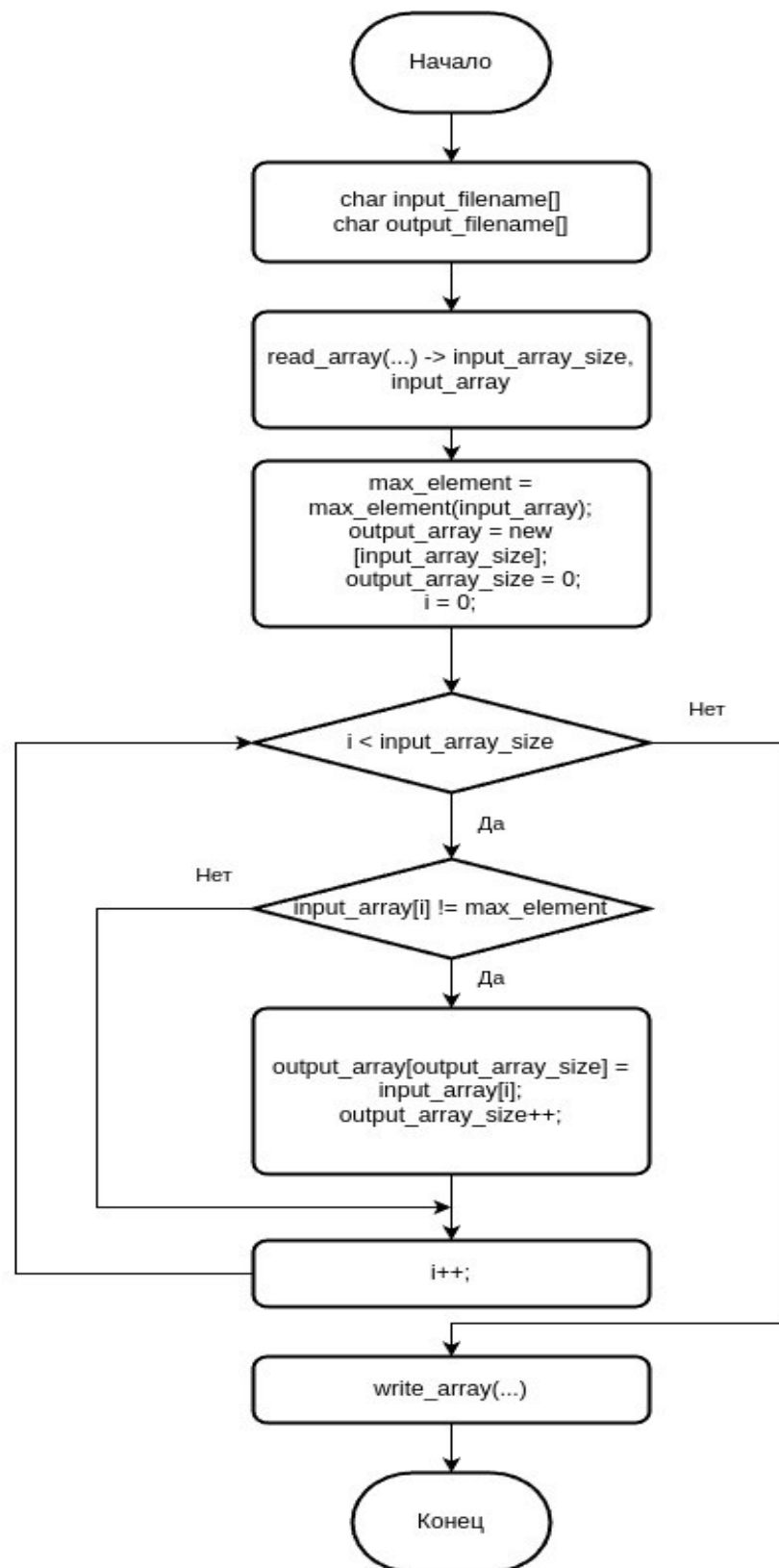


Рисунок 1 – Блок-схема алгоритма

3 Полное описание реализованной функции

3.1 `exclude_max`

Функция `exclude_max` формирует новый массив, исключив все элементы с максимальным значением из входного массива. Принимает следующие аргументы:

1. `const long *input_array`: указатель на входной массив.
2. `size_t input_array_size`: размер входного массива.
3. `long *&output_array`: ссылка на указатель, который будет указывать на выделенный динамический массив, содержащий элементы, исключая максимальные.
4. `size_t &output_array_size`: ссылка на переменную, в которую будет записан размер нового массива.

Работа функции происходит следующим образом:

1. Находит максимальный элемент в массиве `input_array`.
2. Выделяет память для массива `output_array` размером `input_array_size`.
3. Инициализирует `output_array_size` значением 0.
4. Копирует элементы из массива `input_array` в `output_array`, исключая элементы с максимальным значением.
5. Увеличивает `output_array_size` при каждом добавлении элемента в `output_array`.

3.2 `read_array`

Функция `read_array` считывает массив из текстового файла. Принимает следующие аргументы:

1. `const char *filename`: путь к файлу, из которого будет считываться массив.
2. `long *&array`: ссылка на указатель, который будет указывать на выделенный динамический массив, содержащий считанные элементы.
3. `size_t &size`: ссылка на переменную, в которую будет записан размер массива.

Работа функции происходит следующим образом:

1. Открывает файл с именем `filename` для чтения.

2. Если файл не удастся открыть, выводит сообщение об ошибке и завершает программу.
3. Считывает размер массива из файла и записывает его в `size`.
4. Выделяет память для массива `array` размером `size`.
5. Считывает элементы массива из файла в `array`.
6. Закрывает файл.

3.3 write_array

Функция `write_array` записывает массив в текстовый файл. Принимает следующие аргументы:

1. `const char *filename`: путь к файлу, в который будет записан массив.
2. `long *array`: указатель на массив, который будет записан в файл.
3. `size_t &size`: ссылка на переменную, содержащую размер массива.

Работа функции происходит следующим образом:

1. Открывает файл с именем `filename` для записи.
2. Если файл не удастся открыть, выводит сообщение об ошибке и завершает программу.
3. Записывает размер массива `size` в файл.
4. Записывает элементы массива `array` в файл.
5. Закрывает файл.

3.4 main

1. Задаёт пути к входному и выходному файлам.
2. Объявляет указатели и переменные для входного и выходного массивов.
3. Считывает входной массив из файла с помощью `read_array`.
4. Формирует выходной массив, исключая максимальные элементы, с помощью `exclude_max`.
5. Записывает выходной массив в файл с помощью `write_array`.
6. Освобождает выделенную память для массивов.
7. Возвращает 0, сигнализируя об успешном завершении программы.

```
#include <iostream>
#include <fstream>
#include <algorithm>

void read_array(const char *filename, long *&array, size_t &size) {
    std::ifstream input_file(filename);
    if (!input_file.is_open()) {
        std::cerr << "Ошибка открытия файла!" << std::endl;
        exit(1);
    }

    input_file >> size; // Читаем размерность массива

    array = new long[size];

    // Считываем элементы массива из файла
    for (size_t i = 0; i < size; i++)
        input_file >> array[i];

    input_file.close();
}

void write_array(const char *filename, long *array, size_t &size) {
    std::ofstream output_file(filename);
    if (!output_file.is_open()) {
        std::cerr << "Ошибка открытия файла!" << std::endl;
        exit(1);
    }

    // Записываем размерность и элементы выходного массива в файл
    output_file << size << std::endl;
    for (size_t i = 0; i < size; i++)
        output_file << array[i] << " ";

    output_file.close();
}

void exclude_max(const long *input_array, size_t input_array_size, long
*&output_array, size_t &output_array_size) {
    // Находим максимальный элемент в массиве
    long max_element = *std::max_element(input_array, input_array +
input_array_size);

    output_array = new long[input_array_size];
    output_array_size = 0;

    // Проходим по массиву и добавляем элементы в output_array, исключая
    // максимальный элемент
    for (size_t i = 0; i < input_array_size; i++)
        if (input_array[i] != max_element)
            output_array[output_array_size++] = input_array[i];
}
```

```

int main() {
    char input_filename[100] =
"/home/grigorijtomczuk/Desktop/suai/op/lab1.3/input.txt";
    char output_filename[100] =
"/home/grigorijtomczuk/Desktop/suai/op/lab1.3/output.txt";

    long *input_array;
    size_t input_array_size;

    long *output_array;
    size_t output_array_size;

    read_array(input_filename, input_array, input_array_size);
    exclude_max(input_array, input_array_size, output_array,
output_array_size);
    write_array(output_filename, output_array, output_array_size);

    delete[] input_array;
    delete[] output_array;

    return 0;
};

```

5 Несколько тестов работы программы

input.txt ×	
1	7
2	1490273 10293 19028351 19028351 290301 120934 1029
3	
output.txt ×	
1	5
2	1490273 10293 290301 120934 1029

Рисунок 2


```
input.txt x
1 10
2 4 2 4 1 3 0 1 3 4 1
3
output.txt x
1 7
2 2 1 3 0 1 3 1
```

Рисунок 3

```
input.txt x
1 15
2 3425 2345 5 3425 2345 5 6 1 3 3425 6 2345 590 3425 3
3
output.txt x
1 11
2 2345 5 2345 5 6 1 3 6 2345 590 3
```

Рисунок 4