**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**



**МОСКОВСКИЙ ПОЛИТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**ВЫСШАЯ ШКОЛА ПЕЧАТИ И МЕДИАИНДУСТРИИ**

***Институт Принтмедиа и информационных технологий***

***Кафедра Информатики и информационных технологий***

**направление подготовки**

**09.03.02 «Информационные системы и технологии»**

**ЛАБОРАТОРНАЯ РАБОТА № 4-5**

**Дисциплина:** Основы алгоритмизации и программирования.

**Тема:** Алгоритм сортировки «пузырёк»

**Выполнил(а):**

**студент(ка) группы 201-723**

Карпушкин С. Е.

\_\_27.10.2020\_  ***\_\_\_\_\_\_\_\_\_\_\_***

(Дата) (Подпись)

**Проверил:** преп. Хуснулина Д.Р.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  ***\_\_\_\_\_\_\_\_\_\_\_***

(Дата) (Подпись)

**Замечания: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Москва**

**2020**

Оглавление

[Цель 3](#_Toc54702622)

[Задача 3](#_Toc54702623)

[Идея алгоритма 3](#_Toc54702624)

[Блок-схема с использованием элемента модификации 4](#_Toc54702625)

[Блок-схема без элемента модификации 5](#_Toc54702626)

[Исходный код алгоритма с параметрическим циклом 6](#_Toc54702628)

[Исходный код алгоритма с циклом с предусловием 7](#_Toc54702632)

Цель

Получить практические навыки разработки алгоритмов и их программной реализации.

Задача

Необходимо выполнить и оформить описание следующих пунктов:

1. Сформулировать идею алгоритма
2. Выполнить словесное представление алгоритма
3. Выполнить полнить представление алгоритма с помощью блок схем с использованием элемента модификации и без него.
4. Выполнить программную реализацию алгоритмов на языке С с использованием параметрического цикла и цикла с предусловием.

**Идея алгоритма**

Алгоритм основан на повторяющихся проходах по сортируемому массиву. За каждый проход последовательно сравниваются соседние элементы. Если порядок в паре неверный, то происходит обмен значений элементов. При каждом проходе алгоритма по внутреннему циклу, очередной наибольший элемент массива ставится на своё место в конце массива рядом с предыдущим «наибольшим элементом», а наименьший элемент перемещается на одну позицию к началу массива («всплывает» до нужной позиции, как пузырёк в воде — отсюда и название алгоритма). Проходы по массиву повторяются n-1 (где n – размер массива) раз или до тех пор, пока не будет перестановок, т.е. когда массив окажется отсортированным.

**Словесное представление алгоритма**

array – массив, n – длина массива

1. параметр внешнего цикла i = 0
2. если i < n – 1, то п.3, иначе п.9
3. параметр внутреннего цикла j = 0
4. если j < n – 1 – i, то п.5, иначе п.8
5. если array[j] > array[j + 1], то п.6, иначе п.7
6. обмен значениями array[j] и array[j + 1]
7. j++, п.4
8. i++, п.2
9. конец алгоритма

**Блок-схема с использованием элемента модификации**



Рисунок 1 – Блок-схема с использованием элемента модификации

Блок-схема без элемента модификации



Рисунок 2 – Блок-схема без элемента модификации

Исходный код алгоритма с параметрическим циклом

Листинг 1 – исходный код программы “Пузырёк c циклом for”

#include <iostream>

using namespace std;

int main()

{

int i, j, N[] = { 3, 5, 2, 1, 6, 4, 7 }; // Объявление массива и необходимых переменных

int n = sizeof(N) / sizeof(int); // Длина массива

for (int i = 0; i < n - 1; i++) // Внешний цикл

for (int j = 0; j < n - 1 - i; j++) // Внутренний цикл

if (N[j] > N[j+1]) // Если текущий элемент больше следующего, меняем их местами

{

int buf = N[j];

N[j] = N[j + 1];

N[j + 1] = buf;

}

for (int i = 0; i < n; i++) // Вывод упорядоченного массива

printf("%d ", N[i]);

}

Результат работы

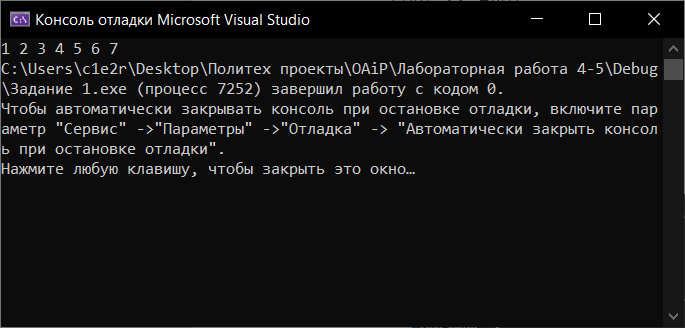


Рисунок 3 – Результат работы программы “Пузырёк c циклом for”

Исходный код алгоритма с циклом с предусловием

Листинг 2 - исходный код программы “Пузырёк c циклом while”

#include <iostream>;

using namespace std;

int main()

{

int i=0, j=0, N[] = { 3, 5, 2, 1, 6, 4, 7 }; // Объявление массива и необходимых переменных

int n = sizeof(N) / sizeof(int); // Длина массива

while (i < n - 1) { // Внешний цикл

while (j < n - 1 - i) { // Внутренний цикл

if (N[j] > N[j + 1]) { // Если текущий элемент больше следующего, меняем их местами

int buf = N[j];

N[j] = N[j + 1];

N[j + 1] = buf;

}

j++; // Увеличиваем j

}

j = 0; // Обнуляем j

i++; // Увеличиваем i

}

for (int i = 0; i < n; i++) // Вывод упорядоченного массива

printf("%d ", N[i]);

}

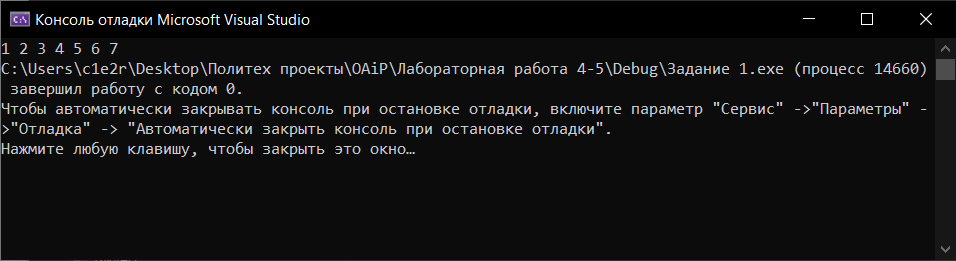
Результат работы

Рисунок 4 – Результат работы программы “Пузырёк c циклом while”