**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 № 8-9**

**Дисциплина:** Основы алгоритмизации и программирования

**Тема:** Алгоритм сортировки «вставками»

**Цель:** Получить практические навыки разработки алгоритмов и их программной реализации.

**Выполнил: студент группы 201-723**

Карпушкин Сергей Евгеньевич

(Фамилия И.О.)



**Дата, подпись** 25.11.2020\_  ***\_\_\_\_\_\_\_\_\_\_\_***

(Дата) (Подпись)

**Проверил: \_\_*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_***

(Фамилия И.О., степень, звание) **(Оценка)**

**Дата, подпись** \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  ***\_\_\_\_\_\_\_\_\_\_\_***

(Дата) (Подпись)

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

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

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

**Москва**

**2020**

Оглавление

[Цель 3](#_Toc57198842)

[Задача 3](#_Toc57198843)

[Идея алгоритма 3](#_Toc57198844)

[Словесное представление алгоритма 3](#_Toc57198845)

[Блок-схема с использованием элемента “решение” 4](#_Toc57198846)

[Блок-схема без элемента “решение” 5](#_Toc57198847)

[Исходный код алгоритма с циклом с предусловием 5](#_Toc57198848)

[Исходный код алгоритма с параметрическим циклом 6](#_Toc57198850)

# 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

Сортируемый массив можно разделить на две части — отсортированная часть и

неотсортированная. В начале сортировки первый элемент массива считается отсортированным, все остальные — не отсортированные. Начиная со второго элемента массива и заканчивая последним, алгоритм вставляет неотсортированный элемент массива в нужную позицию в отсортированной части массива.

# Словесное представление алгоритма

n – длина массива

1 Номер анализ. эл-та равен единице

2 Если номер анализ. эл-та < n, то п. 3, иначе п. 9

3 Запоминаем значение анализ. эл-та

4 Номер текущего эл-та равен номеру анализ. эл-та

5 Если номер текущего элемента > 0 и значение анализируемого элемента < значения элемента, предшествующего текущему, то п. 6, иначе п. 7

6 Значение текущего эл-та равно значению элемента с номером (н.т.э - 1), п. 5

7 В текущий элемент записать значение анализ. эл-та

8 Уменьшить на 1 номер анализ. эл-та, п. 2

9 Конец алгоритма

# Блок-схема с использованием элемента “решение”



Рисунок 1- Блок-схема с использованием элемента “решение”

# Блок-схема без элемента “решение”



Рисунок 2- Блок-схема без элемента “решение”

# Исходный код алгоритма с циклом с предусловием

Листинг 1 – исходный код программы “сортировка вставками с циклом while”

#include <stdio.h>

#include <locale.h>

int main()

{

setlocale(LC\_ALL, "Russian");

int i, j, x;

int array[] = { 5, 0, 2, 4, 1, 3 };

int n = sizeof(array) / sizeof(int); // Объявление массива и необходимых переменных

for (i = 1; i < n; i++) // Начинаем со 2-го элемента массива

{

x = array[i];

j = i;

while (j > 0 && x < array[j-1]) // Пока текущий элемент меньше предыдущего и его индекс не равен 0

Продолжение листинга 1

{

array[j] = array[j-1]; // Перемещаем его влево

j--;

}

array[j] = x;

}

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

{

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

}

}

# Результат работы

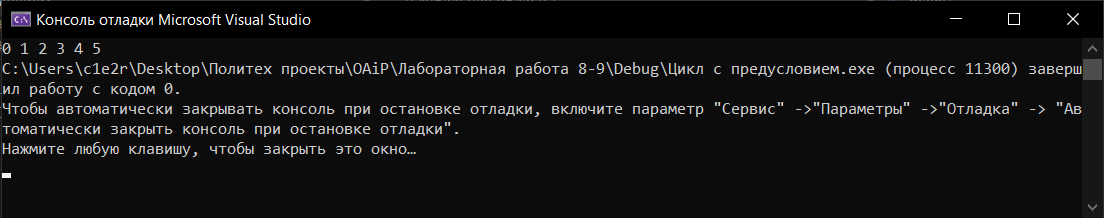


Рисунок 3 - результат работы программы “сортировка вставками с циклом while”

# Исходный код алгоритма с параметрическим циклом

#include <stdio.h>

#include <locale.h>

int main()

{

setlocale(LC\_ALL, "Russian");

int i, j, x;

int array[] = { 5, 0, 2, 4, 1, 3 };

int n = sizeof(array) / sizeof(int); // Объявление массива и необходимых переменных

for (i = 1; i < n; i++) // Начинаем со 2-го элемента массива

{

for (j = i, x = array[i]; (j > 0) && (array[j-1] > x); j--) // Пока текущий элемент меньше предыдущего и его индекс не равен 0

array[j] = array[j-1]; // Перемещаем его влево

array[j] = x;

}

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

{

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

}

}

# Результат работы

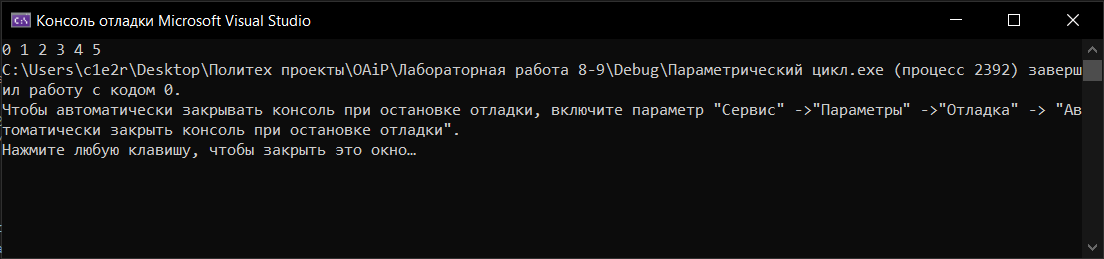


Рисунок 4 - результат работы программы “сортировка вставками с циклом for”