**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 № 6-7**

**Дисциплина:** Основы алгоритмизации и программирования.

**Тема:** Алгоритм сортировки «расческа»

**Выполнила:**

**студентка группы 201-723**

Круглова А.М.

02.11.20  ***\_\_\_\_\_\_\_\_\_\_\_***

(Дата) (Подпись)

**Проверил:** преп. Хуснулина Д.Р.

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

(Дата) (Подпись)

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

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

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

**Москва**

**2020**

**Оглавление**

[Цель 3](#_Toc55261331)

[Постановка задачи 3](#_Toc55261332)

[Идея алгоритма 3](#_Toc55261333)

[Словесное представление алгоритма 4](#_Toc55261334)

[Блок-схема с использованием элемента модификации 5](#_Toc55261335)

[Листинг программы с использованием параметрического цикла 6](#_Toc55261336)

[Результат работы программы 7](#_Toc55261337)

[Блок-схема без использования элемента модификации 8](#_Toc55261338)

[Листинг программы с использованием цикла с предусловием 9](#_Toc55261339)

[Результат работы программы 10](#_Toc55261340)

# Цель:

Получить практические навыки разработке алгоритмов и их программной реализации.

# Постановка задачи:

Необходимо выполнить и оформить описание следующих пунктов:

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

# Идея алгоритма:

Алгоритм является модификацией «пузырька». Отличие алгоритмов состоит в том, что

сравниваются не соседние элементы, а отстоящие друг от друга на определённую

величину, или шаг (назовём его step). Алгоритм реализован с помощью двух циклов.

Окончание внешнего цикла (и алгоритма) происходит тогда, когда step станет меньше 1

На первой итерации расстояние (step) максимально возможное (размер массива – 1), а на

после-дующих итерациях оно изменяется по формуле step /= k (дробная часть

отбрасывается). k – это фактор уменьшения, константа, равная 1.2473309 (при написании

программы можно использовать примерное значение, равное 1.247). Во внутреннем цикле

движение происходит от начала к концу, перемещаясь на step. Если значение текущего

элемента больше, чем значение элемента через step шагов от текущего, то сравниваемые

элементы меняются местами. Условием продолжения цикла является условие i < n – step

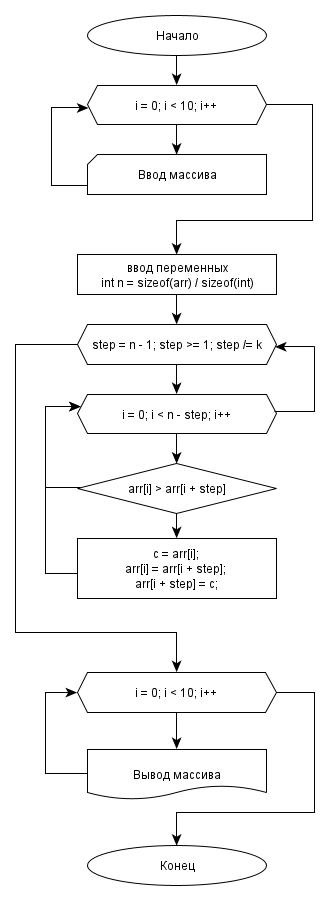
(где i – номер текущего элемента).

Словесное представление алгоритма:

array – массив, n – длина массива, k – фактор уменьшения, равный 1.247, step – шаг

1. расчет шага (step = n – 1)
2. если step >= 1, то п.3, иначе п.10
3. параметр внутреннего цикла i = 0
4. если step >= 1, то п.3, иначе переход к п.10
5. если i < n – step, то п.6, иначе п.9
6. если array[i] > array[i + step], то п.7, иначе п.8
7. перестановка array[i] и array[i + step]
8. i++, п.4
9. step/=k, п.2
10. конец алгоритма

# Блок-схема с использованием элемента модификации:



Нет

Да

Рисунок 1 — Блок-схема 1 программы

Листинг программы с использованием параметрического цикла:

Листинг 1 — Исходный код 1 программы

#include <iostream>

#include <stdlib.h>

int main()

{

setlocale(LC\_ALL, "Russian"); //установка русского языка

int i, step, с;

float k = 1.247; //фактор уменьшения

int arr[10]; // Объявляем массив из 10 элементов

for (i = 0; i < 10; i++) // Вводим значения элементов массива через клавиатуру

{

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

scanf\_s("%d", &arr[i]);

}

int n = sizeof(arr) / sizeof(int); //размер массива

for (step = n - 1; step >= 1; step /= k) //внешний цикл

{

for (i = 0; i < n - step; i++) //внутренний цикл

{

if (arr[i] > arr[i + step]) //условие, если текущий элемент больше другого

{

с = arr[i];

arr[i] = arr[i + step]; //меняем их местами

arr[i + step] = с;

}

}

}

// Выводим отсортированные элементы массива

printf("Отсортированный массив:\n");

for (i = 0; i < 10; i++)

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

printf("\n");

system("pause");

return 0;

}

Результат работы программы:

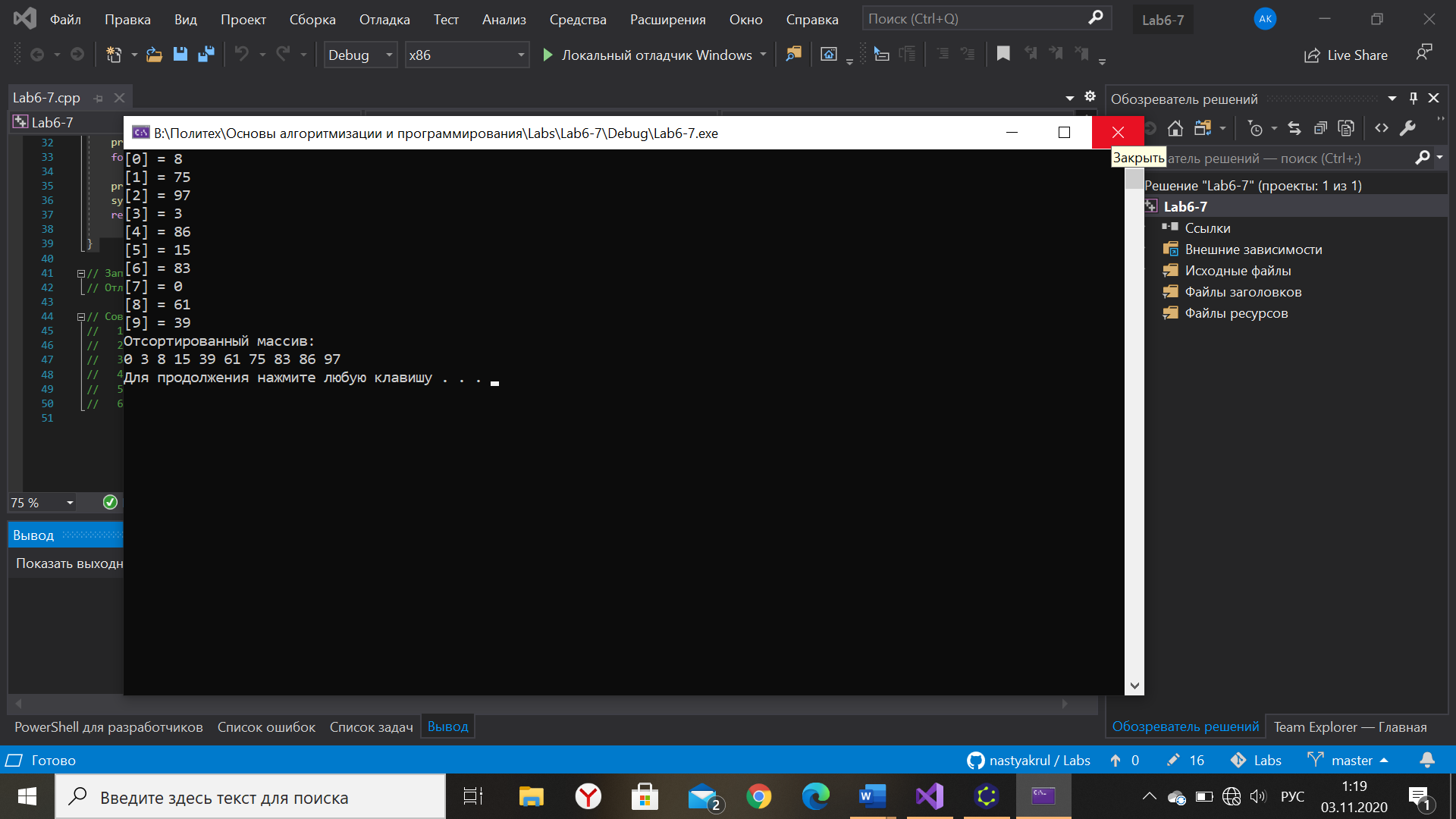
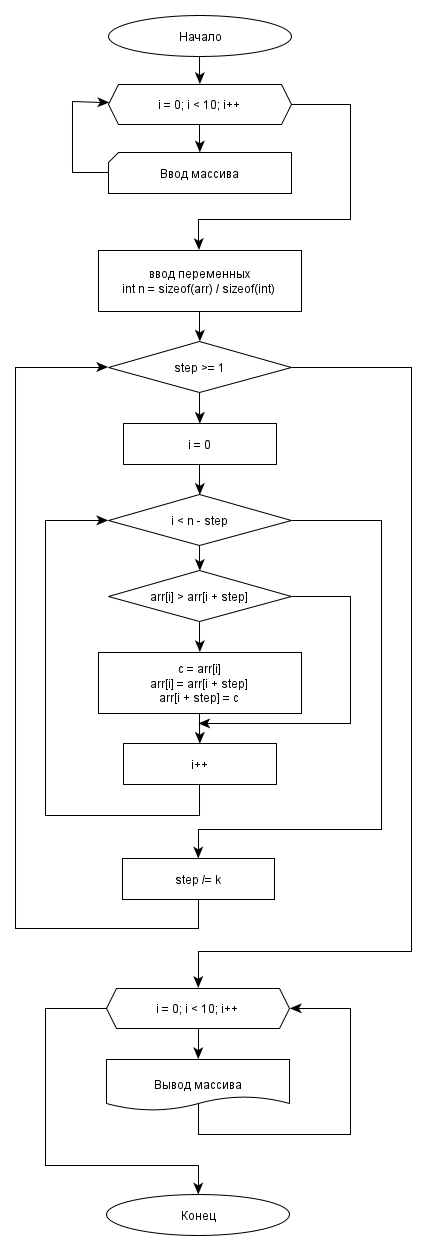


Рисунок 2 — Результат работы 1 программы

Блок-схема без использования элемента модификации:



Нет

Нет

Нет

Да

Да

Да

Рисунок 3 — Блок-схема 2 программы

Листинг программы с использованием цикла с предусловием:

Листинг 2 — Исходный код 2 программы

#include <iostream>

#include <stdlib.h>

int main()

{

setlocale(LC\_ALL, "Russian"); //установка русского языка

int i, c;

float k = 1.247; //фактор уменьшения

int arr[10]; // Объявляем массив из 10 элементов

for (i = 0; i < 10; i++) // Вводим значения элементов массива через клавиатуру

{

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

scanf\_s("%d", &arr[i]);

}

int n = sizeof(arr) / sizeof(int); //размер массива

int step = n - 1;

while (step >= 1) //начало внешнего цикла

{

i = 0;

while (i < n - step) //начало внутреннего цикла

{

if (arr[i] > arr[i + step]) //условие, если текущий элемент больше другого

{

c = arr[i];

arr[i] = arr[i + step]; //меняем местами

arr[i + step] = c;

}

i++;

}

step /= k;

}

// Выводим отсортированные элементы массива

printf("Отсортированный массив:\n");

for (i = 0; i < 10; i++)

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

printf("\n");

system("pause");

return 0;

}

Результат работы программы:

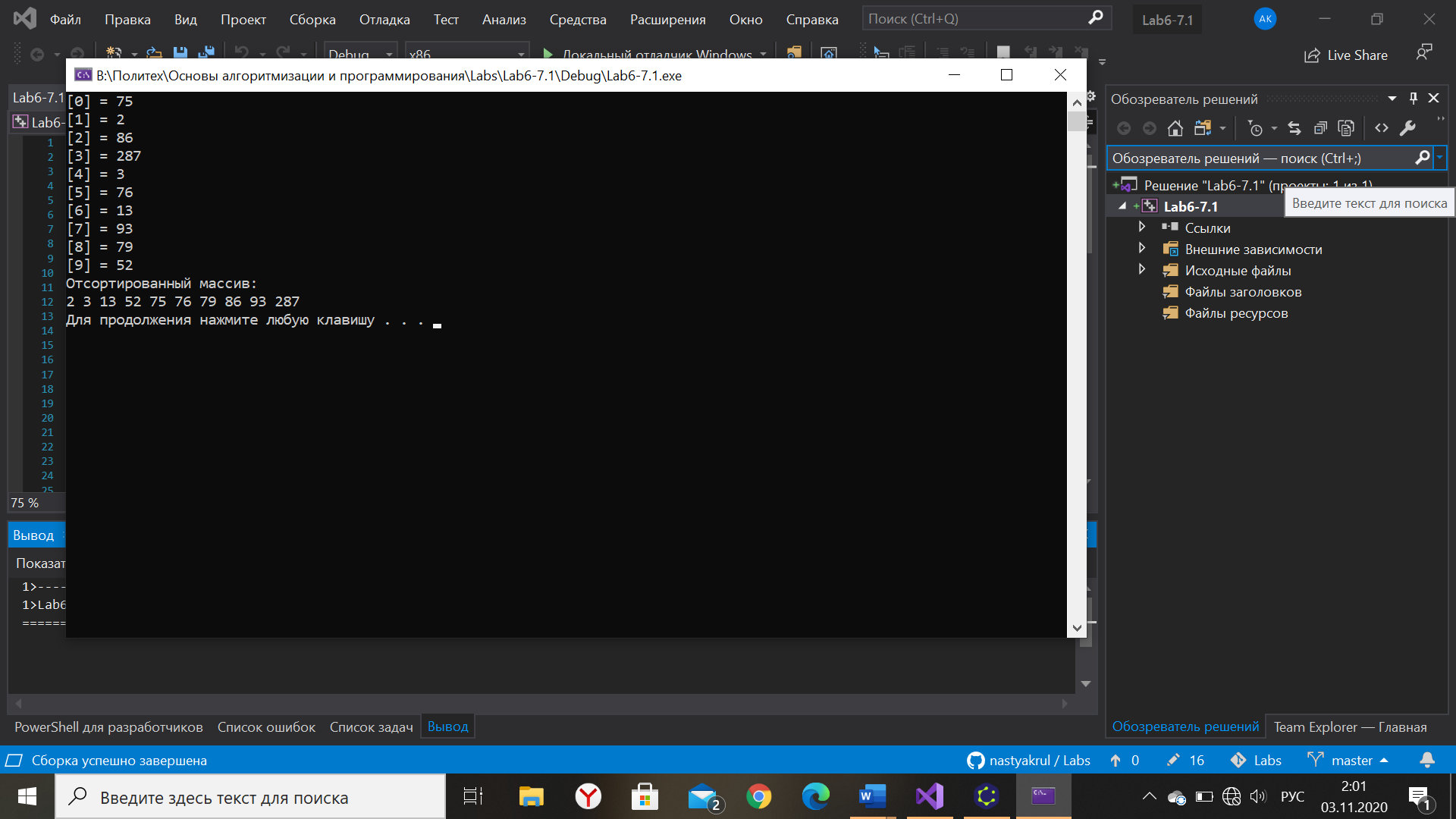


Рисунок 4 — Результат работы 2 программы