**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»

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

**Выполнил: студент группы 201-723**

Карпушкин Сергей Евгеньевич

(Фамилия И.О.)



**Дата, подпись** 03.11.2020\_  ***\_\_\_\_\_\_\_\_\_\_\_***

(Дата) (Подпись)

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

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

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

(Дата) (Подпись)

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

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

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

**Москва**

**2020**

Оглавление

[Цель 3](#_Toc55337993)

[Задача 3](#_Toc55337994)

[Идея алгоритма 3](#_Toc55337995)

[Словесное представление алгоритма 3](#_Toc55337996)

[Блок-схема с использованием элемента модификации 4](#_Toc55337997)

[Блок-схема без элемента модификации 5](#_Toc55337998)

[Исходный код алгоритма с параметрическим циклом 6](#_Toc55337999)

[Исходный код алгоритма с циклом с предусловием 7](#_Toc55338000)

# ****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 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 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****

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 “Расческа с циклом for”

#include <stdio.h>

#include <locale.h>

int main() {

setlocale(LC\_ALL, "Russian");

int array[] = { 1, 4, 5, 0, 3, 2 }; // Объявление массива и необходимых переменных

int n = sizeof(array) / sizeof(int);

float k = 1.247; // фактор уменьшения

for (int step = n - 1; step >= 1; step /= k) // от максимального расстояния между сравниваемыми элментами движемся к 1

for (int i = 0; i < n - step; i++)

if (array[i] > array[i + step]) // сравниваем текущий элемент и находящийся от него на расстоянии step

{

int temp = array[i]; // если текущий элемент больше, меняем их местами

array[i] = array[i + step];

array[i + step] = temp;

}

for (int i = 0; i < n; i++) // вывод отсортированного массива

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

}

**Результат работы**

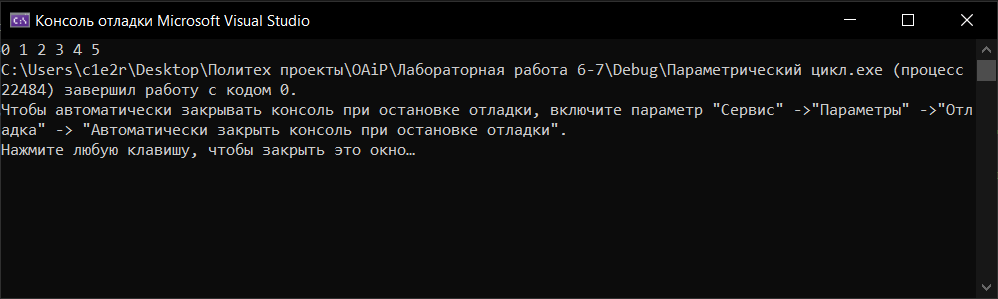


Рисунок 3 – Результат работы программы “ Расческа с циклом for”

# ****Исходный код алгоритма с циклом с предусловием****

Листинг 2 – исходный код программы “Расческа с циклом while”

#include <stdio.h>

#include <locale.h>

int main() {

setlocale(LC\_ALL, "Russian");

int array[] = { 1, 4, 5, 0, 3, 2 }; // Объявление массива и переменных

int n = sizeof(array) / sizeof(int);

float k = 1.247; // фактор уменьшения

int step = n - 1;

while (step >= 1) // пока шаг больше 0

{

int i = 0; // начинаем с 0-ого элемента

while (i < n - step)

{

if (array[i] > array[i + step]) // если текущий элемент больше, меняем местами

{

int temp = array[i];

array[i] = array[i + step];

array[i + step] = temp;

}

i++;

}

step /= k; // уменьшаем расстояние на фактор уменьшения

}

for (int i = 0; i < n; i++) // вывод отсортированного массива

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

}

**Результат работы**

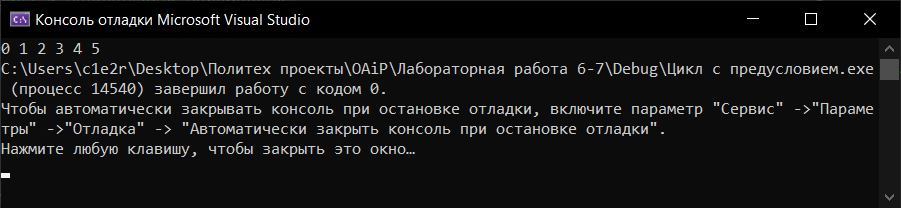
****

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