



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
Задача.....	3
Идея алгоритма.....	3
Словесное представление алгоритма	3
Блок-схема с использованием элемента модификации.....	4
Блок-схема без элемента модификации	5
Исходный код алгоритма с параметрическим циклом.....	6
Исходный код алгоритма с циклом с предусловием.....	7

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 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 последующих 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. Условие 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 $\text{step} \geq 1$, то п.3, иначе п.10
- 3 параметр внутреннего цикла $i = 0$
- 4 если $\text{step} \geq 1$, то п.3, иначе переход к п.10
- 5 если $i < n - \text{step}$, то п.6, иначе п.9
- 6 если $\text{array}[i] > \text{array}[i + \text{step}]$, то п.7, иначе п.8
- 7 перестановка $\text{array}[i]$ и $\text{array}[i + \text{step}]$
- 8 $i++$, п.4
- 9 $\text{step} /= k$, п.2
- 10 конец алгоритма

Блок-схема с использованием элемента модификации

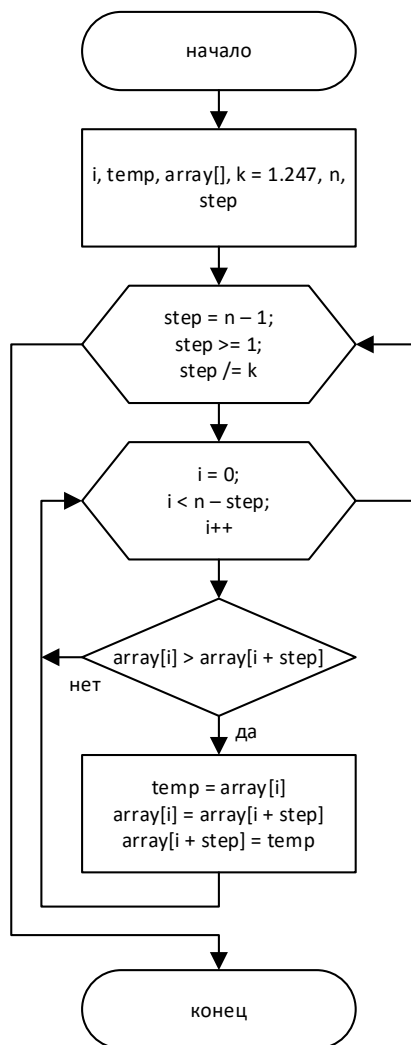


Рисунок 1- Блок-схема с использованием элемента модификации

Блок-схема без элемента модификации

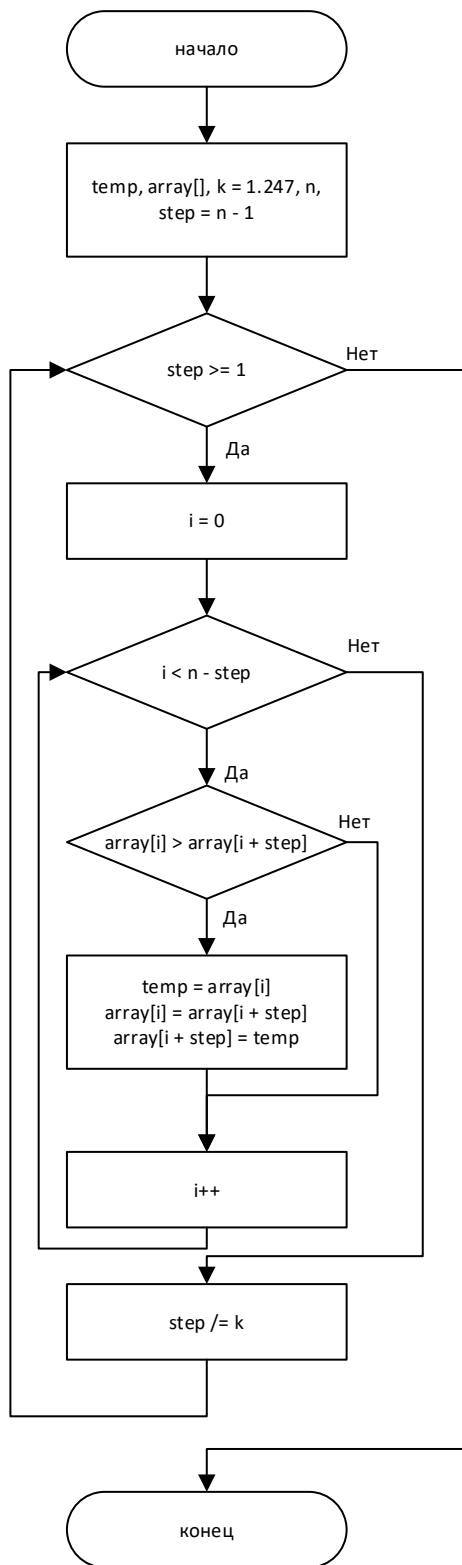


Рисунок 2 – Блок-схема без элемента модификации

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 элементами 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

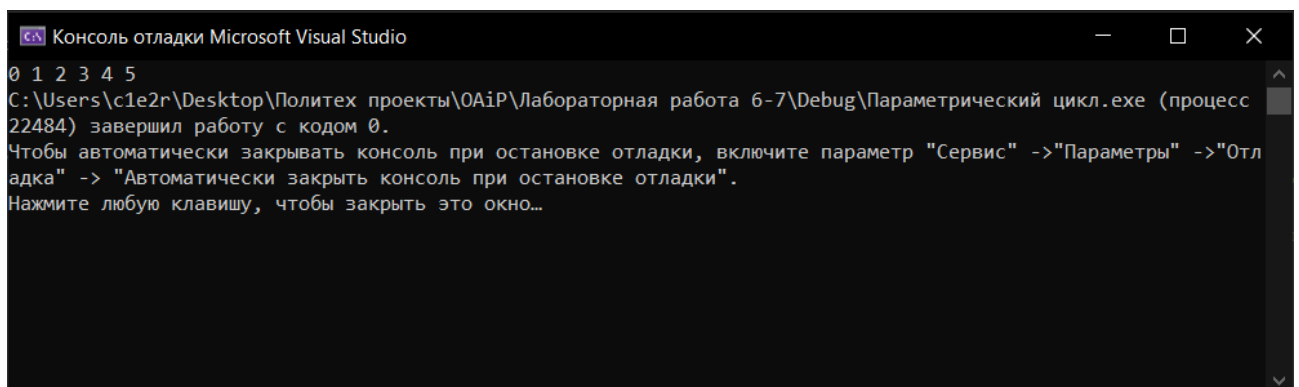


Рисунок 3 – Результат работы программы “Расческа с циклом for”

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

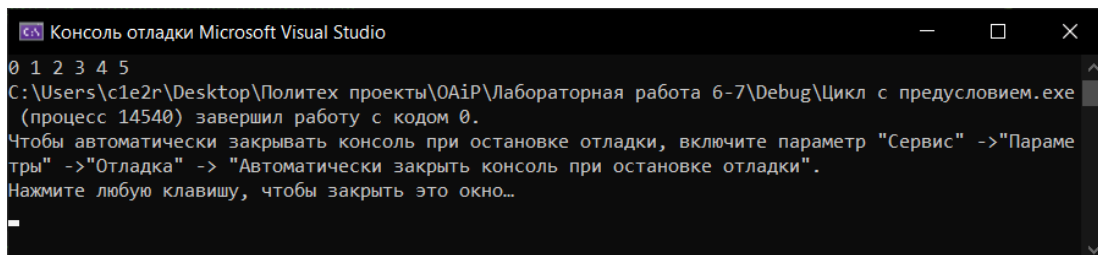


Рисунок 4 – Результат работы программы “Расческа с циклом while”