



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 № 10-11

Дисциплина: Основы алгоритмизации и программирования

Тема: Алгоритм сортировки “Шелла”

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
Задача.....	3
Идея алгоритма.....	3
Словесное представление алгоритма	3
Блок-схема с использованием элемента “модификация”	4
Блок-схема с использованием элемента “решение”	5
Исходный код алгоритма “Сортировка Шелла с параметрическим циклом”	6
Исходный код алгоритма “Сортировка Шелла 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 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 сортирует элементы, отстоящие друг от друга на некотором расстоянии. Затем сортировка повторяется при меньших значениях шага, и в конце процесс сортировки Шелла завершается при шаге, равном 1 (а именно обычной сортировкой вставками). Шелл предложил такую последовательность размера шага: $N/2, N/4, N/8 \dots$, где N – количество элементов в сортируемом массиве.

Словесное представление алгоритма

n – длина массива, d – шаг

- 1 Рассчитываем начальное значение шага: $d = n / 2$
- 2 Если $d > 0$, то п.3, иначе п. 14
- 3 Номер анализ. эл-та = d
- 4 Если номер анализ. элемента $< n$, то п.5, иначе п.13
- 5 Запоминаем значение анализ. элемента
- 6 Номер текущего элемента = номеру анализ. элемента
- 7 Если номер текущего элемента $\geq d$, то п.8, иначе п.11
- 8 Если значение текущего элемента $<$ значение элемента с номером (текущего элемента – d), то п.9, иначе п.11
- 9 Значение т.э. = значение эл-та с номером (т.э – d)
- 10 Номер т.э = номер (т.э – d), п. 7
- 11 Значение т.э = значение анализ. элемента
- 12 $i++$, п. 4
- 13 $d /= 2$, п. 2
- 14 Конец алгоритма

Блок-схема с использованием элемента “модификация”

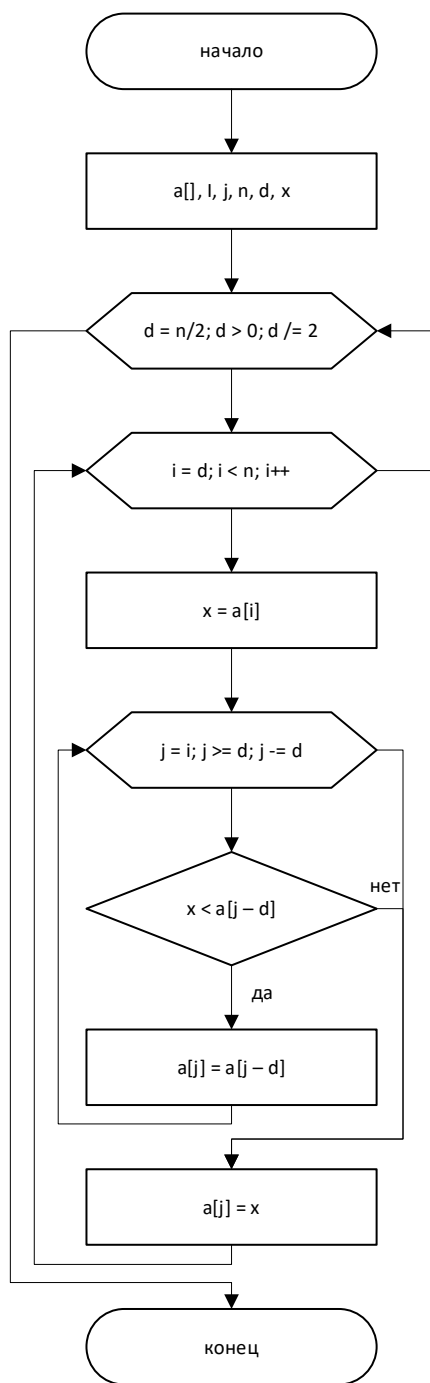


Рисунок 1 – Блок-схема с использованием элемента “модификация”

Блок-схема с использованием элемента “решение”

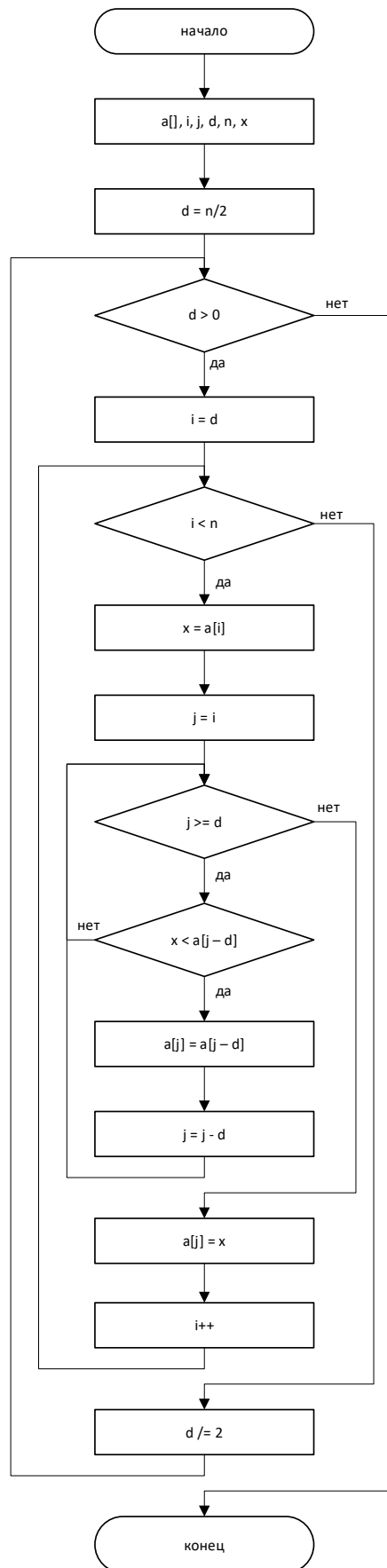


Рисунок 2 - Блок-схема с использованием элемента “решение”

Исходный код алгоритма “Сортировка Шелла с параметрическим циклом”

Листинг 1 - Исходный код программы “сортировка Шелла с параметрическим циклом”

```
#include <stdio.h>
#include <locale.h>
int main()
{
    setlocale(LC_ALL, "Russian");
    int i, j, x, d;
    int array[] = { 5, 0, 2, 4, 1, 3 };
    int n = sizeof(array) / sizeof(int); // Объявление массива и необходимых переменных

    for (d = n / 2; d > 0; d /= 2) // Сравниваем элементы на расстоянии шага
        for (i = d; i < n; i++)
        {
            x = array[i];
            for (j = i; j >= d; j -= d)
                if (x < array[j - d]) // Если порядок неверный, меняем местами, затем уменьшаем шаг
                    array[j] = array[j - d];
            else
                break;
            array[j] = x;
        }

    for (i = 0; i < n; i++) // Вывод упорядоченного массива
    {
        printf("%d ", array[i]);
    }
}
```

Результат работы

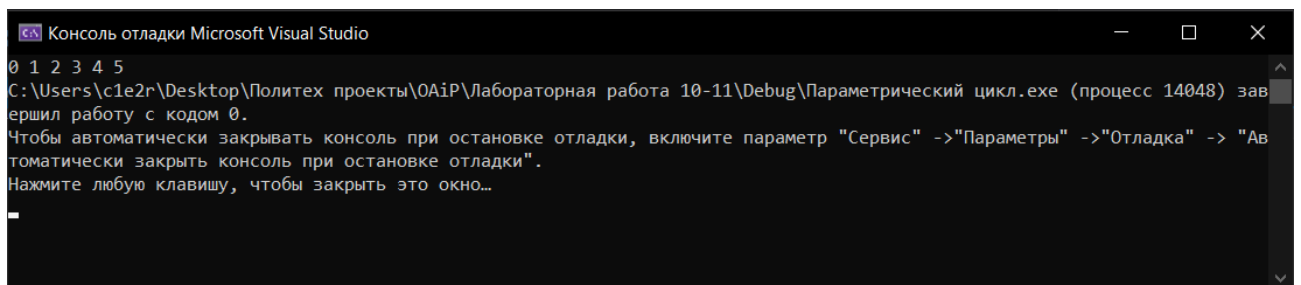


Рисунок 3 – Результат работы программы “сортировка Шелла с параметрическим циклом”

Исходный код алгоритма “Сортировка Шелла с циклом с предусловием”

Листинг 2 - Исходный код программы “сортировка Шелла с циклом с предусловием”

```
#include <stdio.h>
#include <locale.h>
int main()
{
    setlocale(LC_ALL, "Russian");
    int i, j, x, d;
    int array[] = { 5, 0, 2, 4, 1, 3 };
    int n = sizeof(array) / sizeof(int);
    d = n / 2; // Объявление массива и необходимых переменных

    while (d > 0) // Пока шаг больше нуля
    {
        i = d;
        while (i < n)
        {
            x = array[i];
            j = i;
            while (j >= d)
                if (x < array[j - d]) // Если порядок неверный, меняем
местами
                {
                    array[j] = array[j - d];
                    j = j - d;
                }
            else
                break;
            array[j] = x;
            i++;
        }
        d = d / 2; // Уменьшаем шаг
    }

    for (i = 0; i < n; i++) // Вывод упорядоченного массива
    {
        printf("%d ", array[i]);
    }
}
```

Результат работы

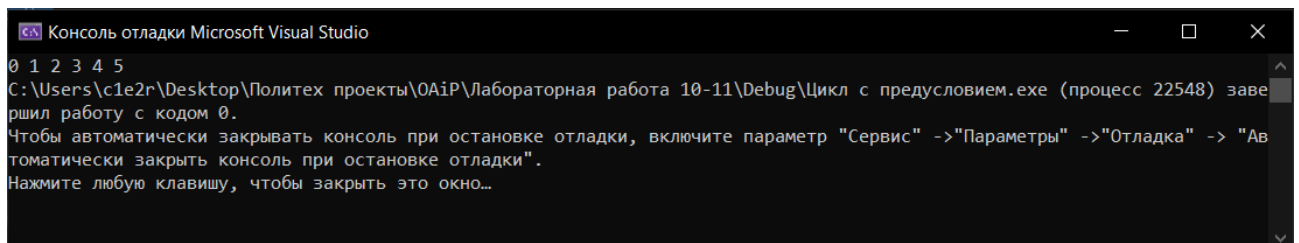


Рисунок 4 – Результат работы программы “сортировка Шелла с циклом с предусловием”