



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 № 14-15

Дисциплина: Основы алгоритмизации и программирования

Тема: Алгоритм сортировки «гномья»

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

Выполнил: студент группы 201-723

Карпушкин Сергей Евгеньевич
(Фамилия И.О.)

Дата, подпись 18.12.2020_

(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 “решение”	4
Исходный код программы “Сортировка гномья”	5
Результат работы	6

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 C с использованием параметрического цикла и цикла с предусловием.

Идея алгоритма

Идея алгоритма очень проста. Пусть имеется массив A размером N, тогда сортировка выбором сводится к следующему:

- Смотрим на текущий и предыдущий элемент массива:
 - если они в правильном порядке, шагаем на один элемент вперед,
 - иначе меняем их местами и шагаем на один элемент назад.
- Граничные условия:
 - если нет предыдущего элемента, шагаем вперёд;
 - если нет следующего элемента, стоп.

Это оптимизированная версия с использованием переменной j, чтобы разрешить прыжок вперёд туда, где он остановился до движения влево, избегая лишних итераций и сравнений.

Словесное представление алгоритма

arr – массив, N- длина массива, i,j- индексы массивов, min – индекс локального минимума

- 1 Сортировка начинается со второго и третьего элементов $i=1, j=2$;
- 2 Если $i < N$, то к пункту 3, иначе к пункту 9
- 3 если $arr[i - 1] > arr[i]$, то к пункту 4, иначе к пункту 7
- 4 Меняем местами значения $arr[i]$ и $arr[i - 1]$
- 5 Шагаем на один элемент назад $i--$
- 6 Если $i > 0$, то к пункту 2(используя оператор continue), иначе к пункту 7
- 7 $i = j++$
- 8 К пункту 2
- 9 Конец алгоритма

Блок-схема с использованием элемента “решение”

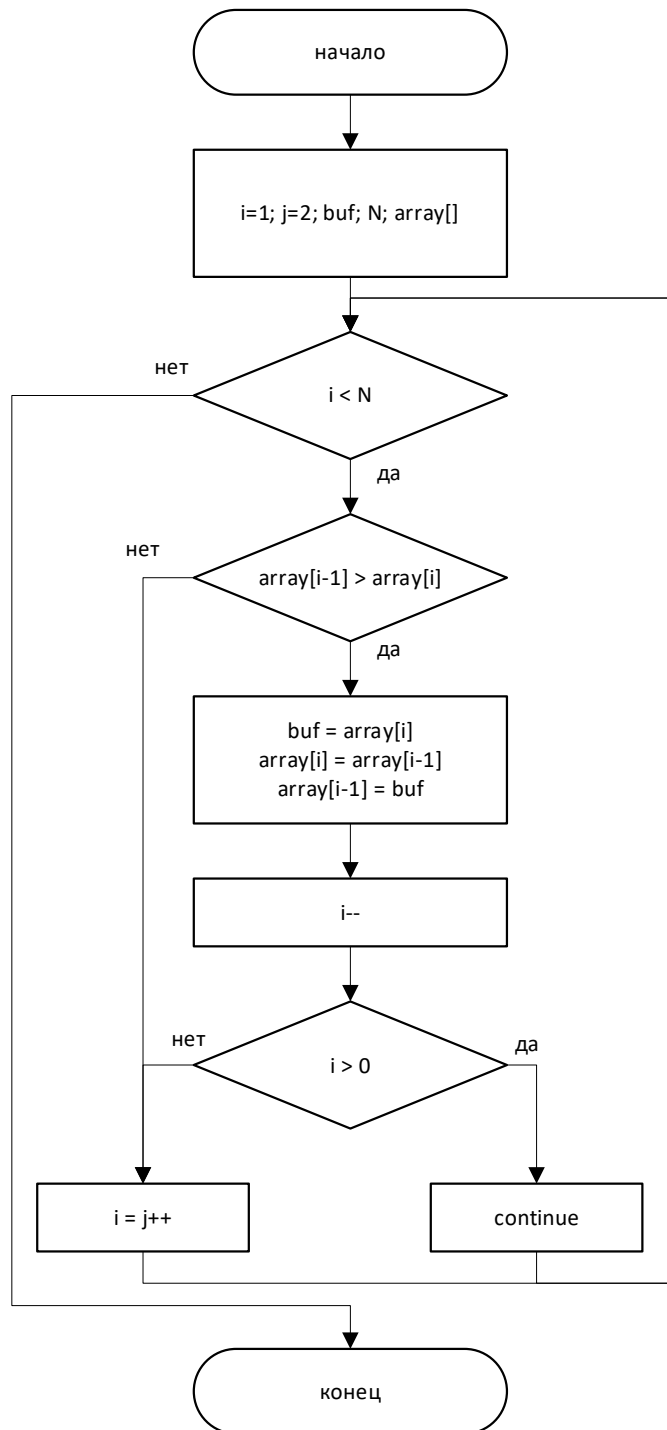


Рисунок 1 - Блок-схема с использованием элемента “решение”

Блок-схема с использованием элемента “модификация”

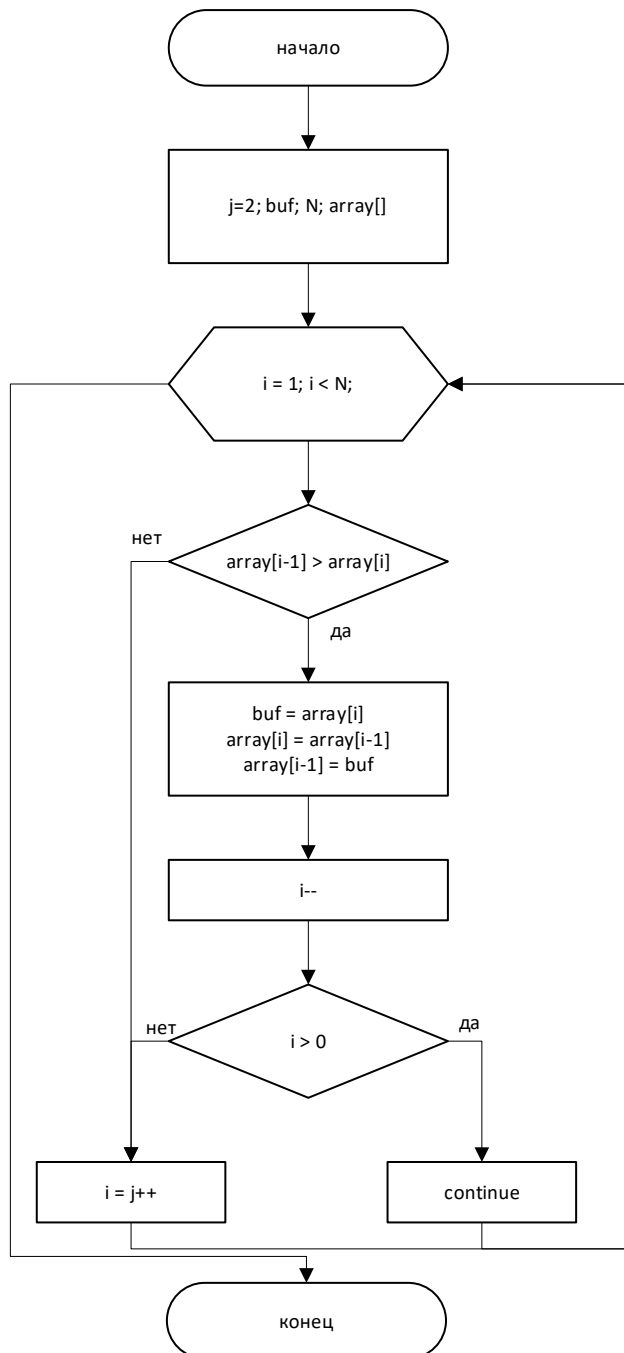


Рисунок 2 - Блок-схема с использованием элемента “модификация”

Исходный код программы “Сортировка гномья с циклом while”

Листинг 1 - Исходный код программы “Сортировка гномья с циклом while”

```
#include <stdio.h>

int main()
{
    int i = 1, j = 2, buf, N = 7; // Объявление необходимых переменных
    int arr[] = { 6, 4, 1, 5, 3, 7, 2 }; // Объявление массива

    while (i < N) { // Двигаемся по всему массиву
        if (arr[i - 1] > arr[i]) { // Если предыдущий элемент больше
текущего
            buf = arr[i]; // Меняем их местами
            arr[i] = arr[i - 1];
            arr[i - 1] = buf;
            i--; // Шагаем на один элемент назад
            if (i > 0) continue;
        }
        i = j++; // Возвращаемся

    }

    for (int i = 0; i < N; i++) // Вывод упорядоченного массива
        printf("%d ", arr[i]);
}
```

Результат работы

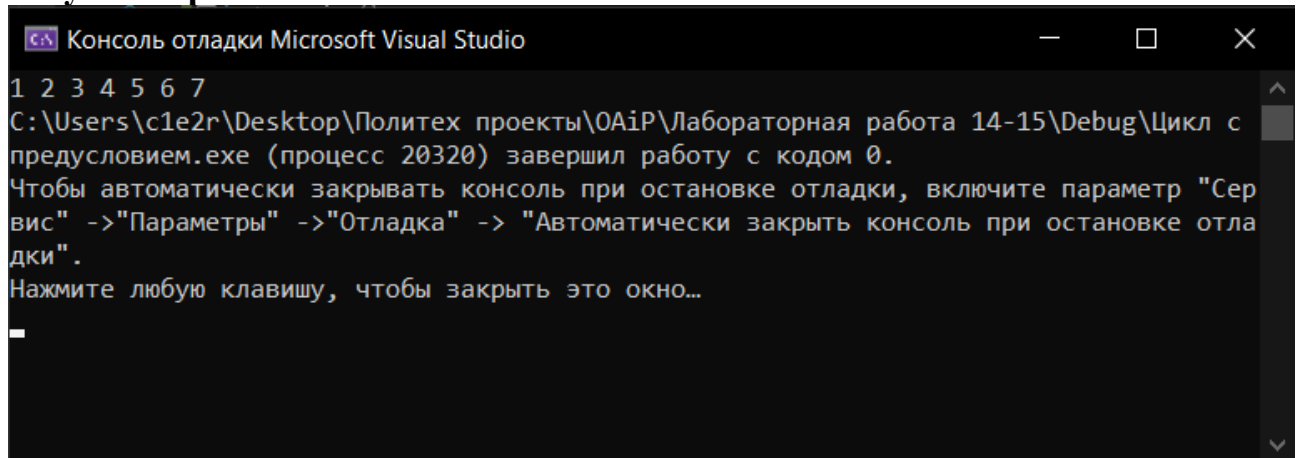


Рисунок 3 – результат работы программы “Сортировка гномья”

Исходный код программы “Сортировка гномья с циклом for”

Листинг 2 - Исходный код программы “Сортировка гномья с циклом for”

```
#include <stdio.h>

int main()
{
    int i = 1, j = 2, buf, N = 7; // Объявление необходимых переменных
    int arr[] = { 6, 4, 1, 5, 3, 7, 2 }; // Объявление массива

    for (i = 1; i < N;)
    {
        if (arr[i - 1] > arr[i]) { // Если предыдущий элемент больше
текущего
            buf = arr[i]; // Меняем их местами
            arr[i] = arr[i - 1];
            arr[i - 1] = buf;
            i--; // Шагаем на один элемент назад
            if (i > 0) continue;
        }
        i = j++;
    }

    for (int i = 0; i < N; i++) // Вывод упорядоченного массива
        printf("%d ", arr[i]);
}
```

Результат работы

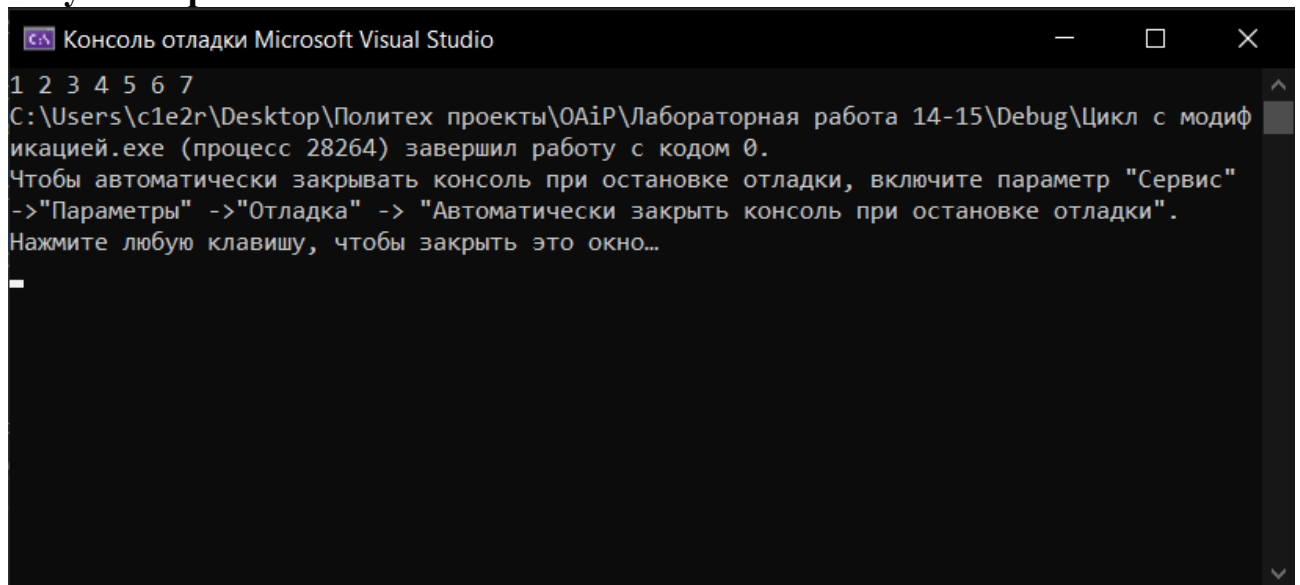


Рисунок 4 – результат работы программы “Сортировка гномья с циклом for”