



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 № 12-13

Дисциплина: Основы алгоритмизации и программирования

Тема: Алгоритм сортировки “выбором”

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

Выполнил: студент группы 201-723

Карпушкин Сергей Евгеньевич
(Фамилия И.О.)

Дата, подпись 17.12.2020_

(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 “модификация”	4
Блок-схема с использованием элемента “решение”	5
Исходный код программы “Сортировка вставками с параметрическим циклом”	6
Исходный код программы “Сортировка вставками с циклом while”	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 С с использованием параметрического цикла и цикла с предусловием.

Идея алгоритма

Пусть имеется массив A размером N , тогда сортировка выбором сводится к следующему:

Берем первый элемент последовательности $A[i]$, здесь i – номер элемента, для первого i равен 1. Находим минимальный (максимальный) элемент последовательности и запоминаем его номер. Если номер первого элемента и номер найденного элемента не совпадают, тогда два этих элемента обмениваются значениями, иначе никаких манипуляций не происходит. Увеличиваем i на 1 и продолжаем сортировку оставшейся части массива. С каждым последующим шагом размер подмассива, с которым работает алгоритм, уменьшается.

Словесное представление алгоритма

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

- 1 Сортировка начинается с первого элемента $i=0$
- 2 Если $i < N - 1$, то п. 3, иначе к пункту 12
- 3 $min = i, j = i + 1$
- 4 Если $j < N$, то к пункту , иначе к пункту 9
- 5 Ищем локальный минимум. Если $array[j] < array[min]$, то к пункту 6, иначе к пункту 7
- 6 Запоминаем новый индекс ($min = j$)
- 7 $j++$
- 8 К пункту 4.
- 9 Обмен значениями $array[i]$ и $array[min]$.
- 10 $i++$
- 11 К пункту 2
- 12 Конец алгоритма

Блок-схема с использованием элемента “модификация”

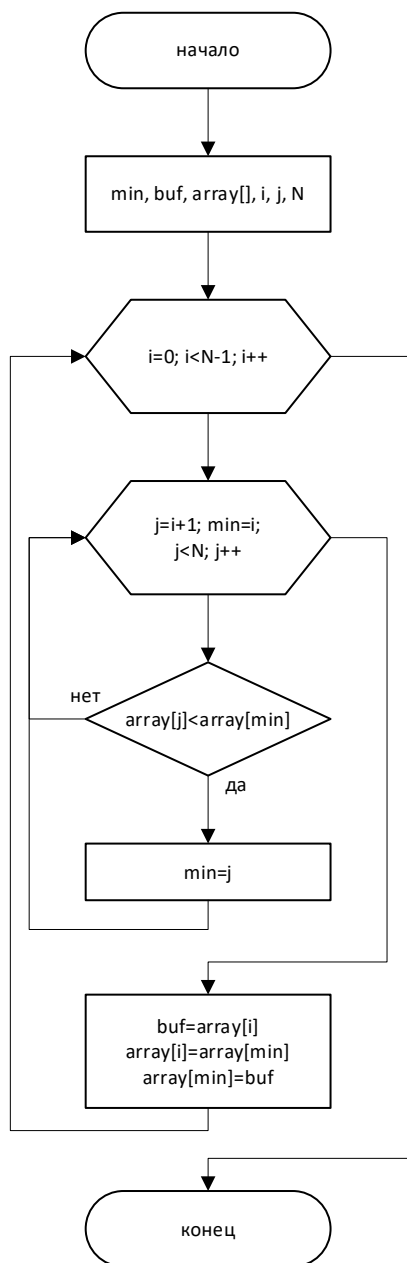


Рисунок 1 – Блок-схема с использованием элемента “модификация”

Блок-схема с использованием элемента “решение”

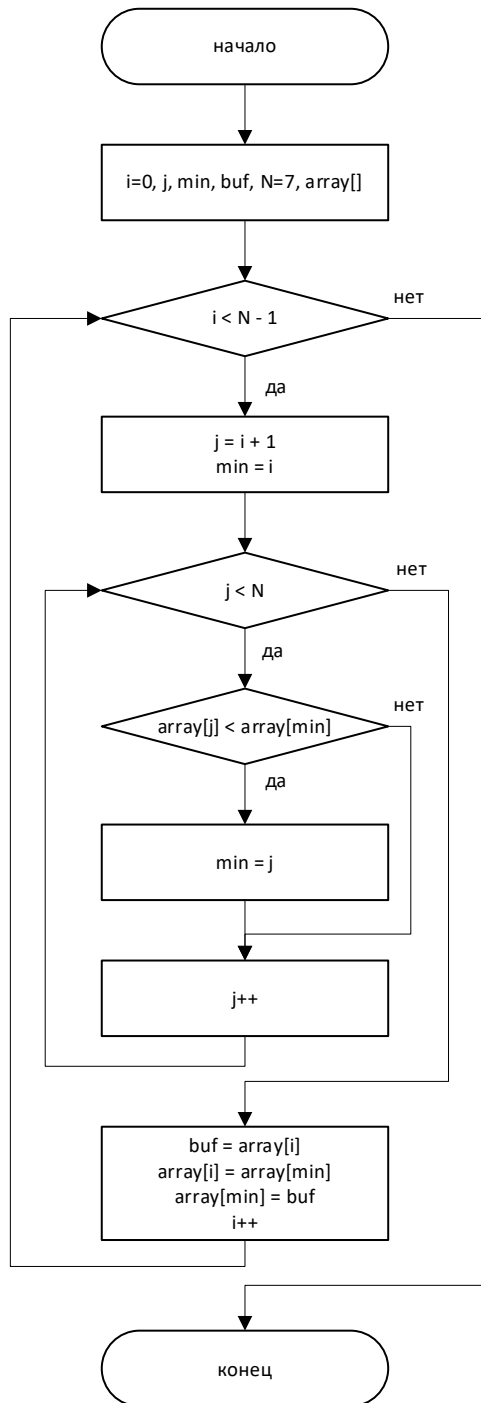


Рисунок 2 - Блок-схема с использованием элемента “решение”

Исходный код программы “Сортировка вставками с параметрическим циклом”

Листинг 1 - Исходный код программы “Сортировка вставками с параметрическим циклом”

```
#include <stdio.h>

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

    for (i = 0; i < N - 1; i++) // Движемся по всему массиву, кроме последнего
        элемента
        {
            for (j = i + 1, min = i; j < N; j++) // Находим локальный максимум
                if (array[j] < array[min])
                    min = j;

            buf = array[i]; // Меняем местами сравниваемое число и локальный
            максимум
            array[i] = array[min];
            array[min] = buf;
        }

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

Результат работы

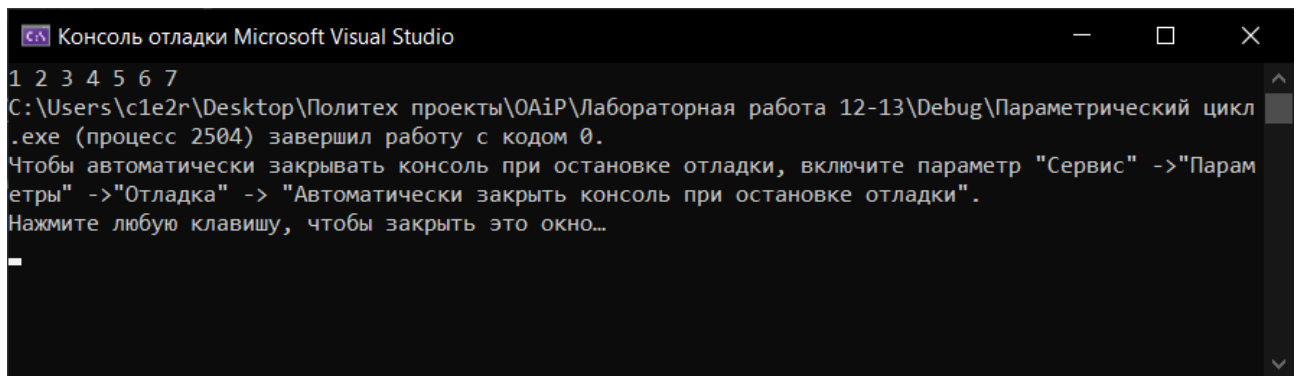


Рисунок 3 – Результат работы программы “Сортировка вставками с параметрическим циклом”

Исходный код программы “Сортировка вставками с циклом с предусловием”

Листинг 2 - Исходный код программы “Сортировка вставками с циклом с предусловием”

```
#include <stdio.h>

int main()
{
```

```

int i=0, j, min, buf, N = 7; // Объявление необходимых переменных
int array[] = { 6, 4, 1, 5, 3, 7, 2 }; // Объявление массива

while (i < N - 1) // Движемся по всему массиву, кроме последнего элемента
{
    j = i + 1;
    min = i;
    while (j < N) // Находим локальный максимум
    {
        if (array[j] < array[min])
            min = j;
        j++;
    }
    buf = array[i]; // Меняем местами сравниваемое число и локальный
    максимум
    array[i] = array[min];
    array[min] = buf;
    i++;
}

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

```

Результат работы

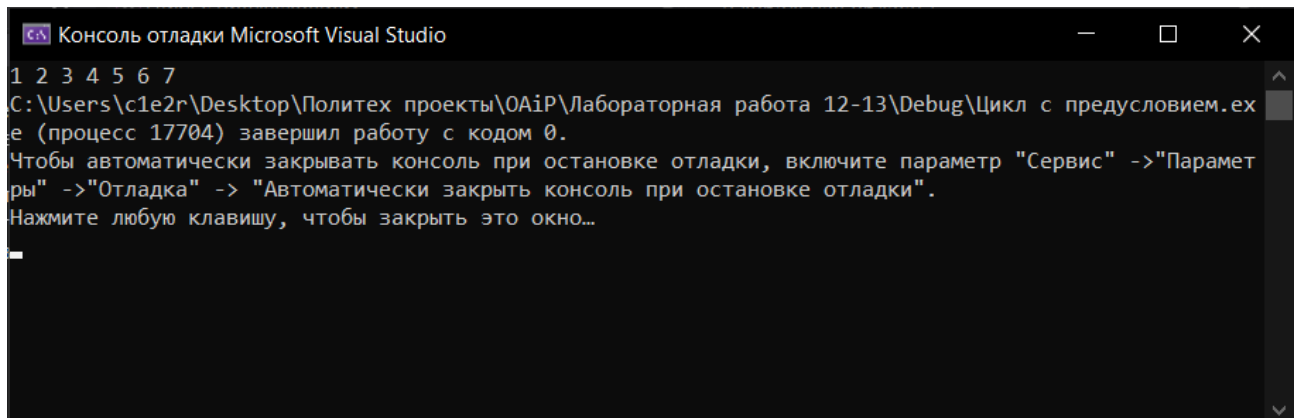


Рисунок 4 – Результат работы программы “Сортировка вставками с циклом с предусловием”