

Лабораторная работа №11.

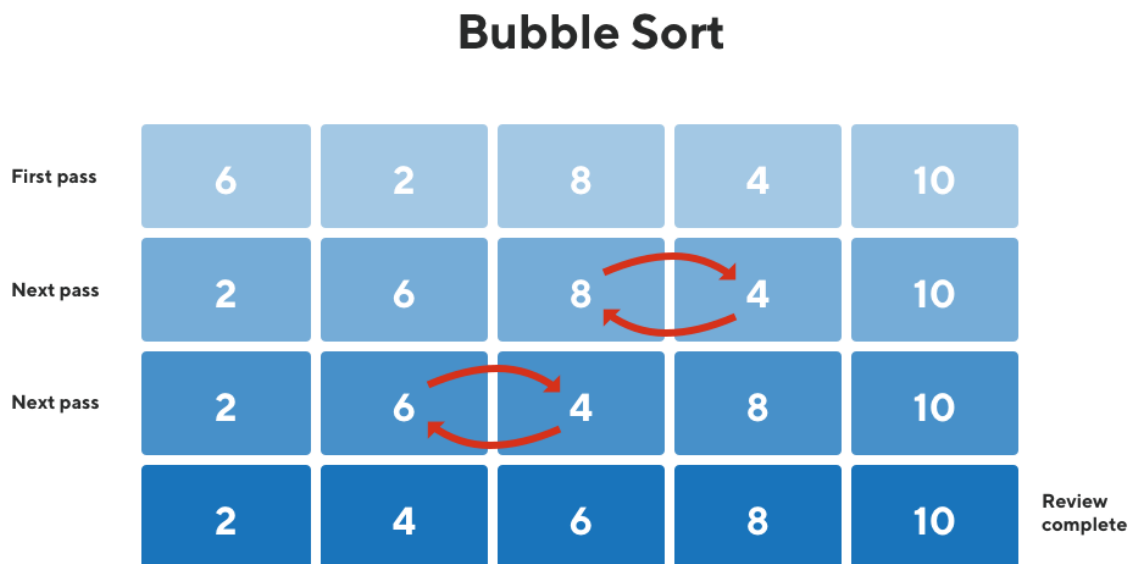
Итерационные циклические вычислительные процессы с управлением по индексу и функции.

Цель: средствами языка C научиться работать с итерационными циклическими вычислительными процессами с управлением по индексу и функции и решить поставленные задачи.

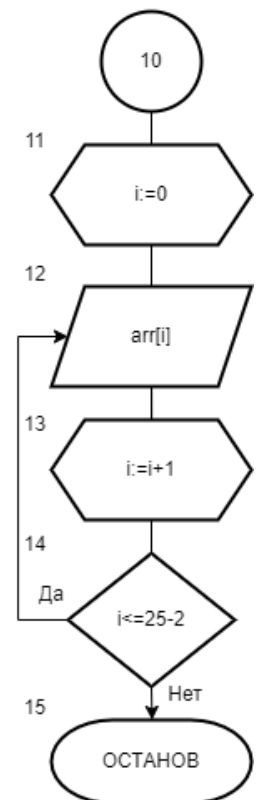
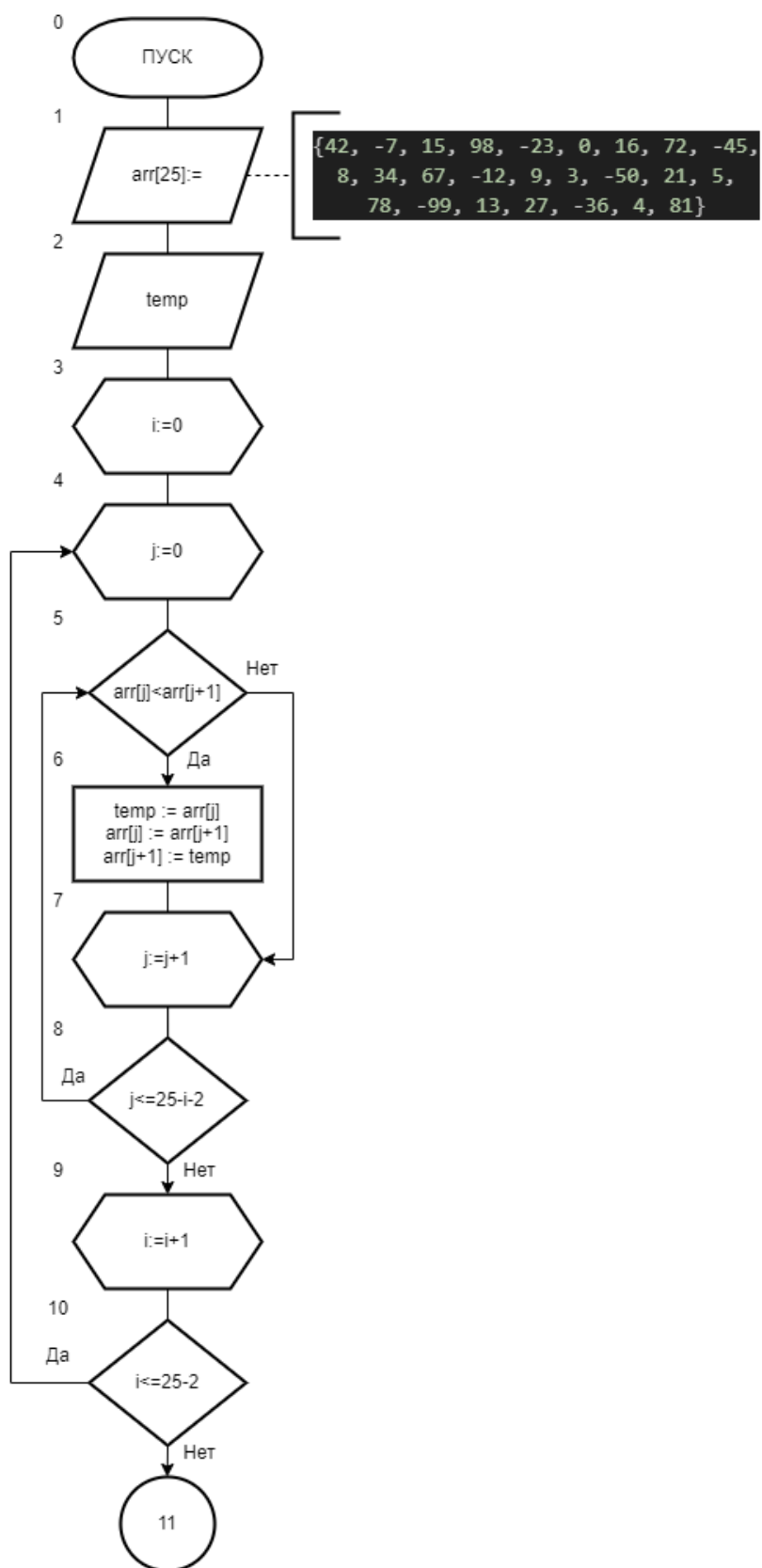
Оборудование: ПК, Visual Studio Code

Задача 1: упорядочить одномерный массив методом пузырька.

Математическая модель:



Блок схема:



Список идентификаторов:

Имя	Тип	Смысл
i	int	Параметр цикла
j	int	Параметр цикла
Arr[]	Int	Массив

Temp	int	Временная переменная
------	-----	----------------------

Код программы:

```
#include <stdio.h>
#include <math.h>

int main(void){

int arr[25] = {42, -7, 15, 98, -23, 0, 16, 72, -45, 8, 34, 67, -12, 9, 3, -50, 21, 5, 78, -99, 13, 27, -36, 4, 81};
int temp;

for (int i = 0; i <= 25 - 2; i++)
{
    for (int j = 0; j <= 25 - i - 2; j++)
    {
        if (arr[j] < arr[j+1]){
            temp = arr[j];
            arr[j] = arr[j+1];
            arr[j+1] = temp;
        }
    }
}

for (int i = 0; i <= 25 - 1; i++)
{
    printf("%d ", arr[i]);
}
}
```

Результат выполнения программы:

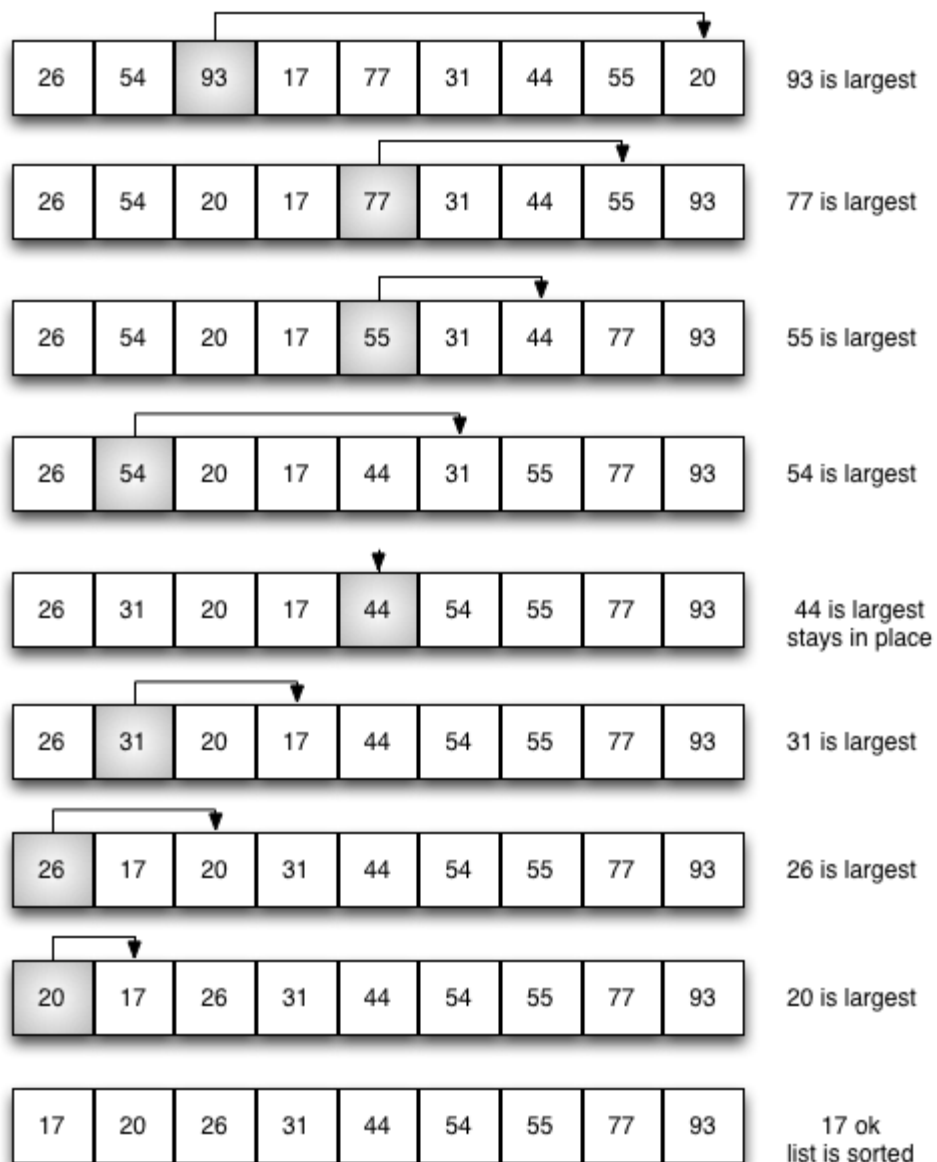
```
98 81 78 72 67 42 34 27 21 16 15 13 9 8 5 4 3 0 -7 -12 -23 -36 -45 -50 -99
```

Анализ вычислений:

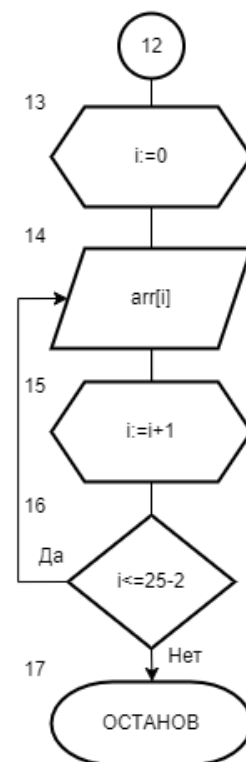
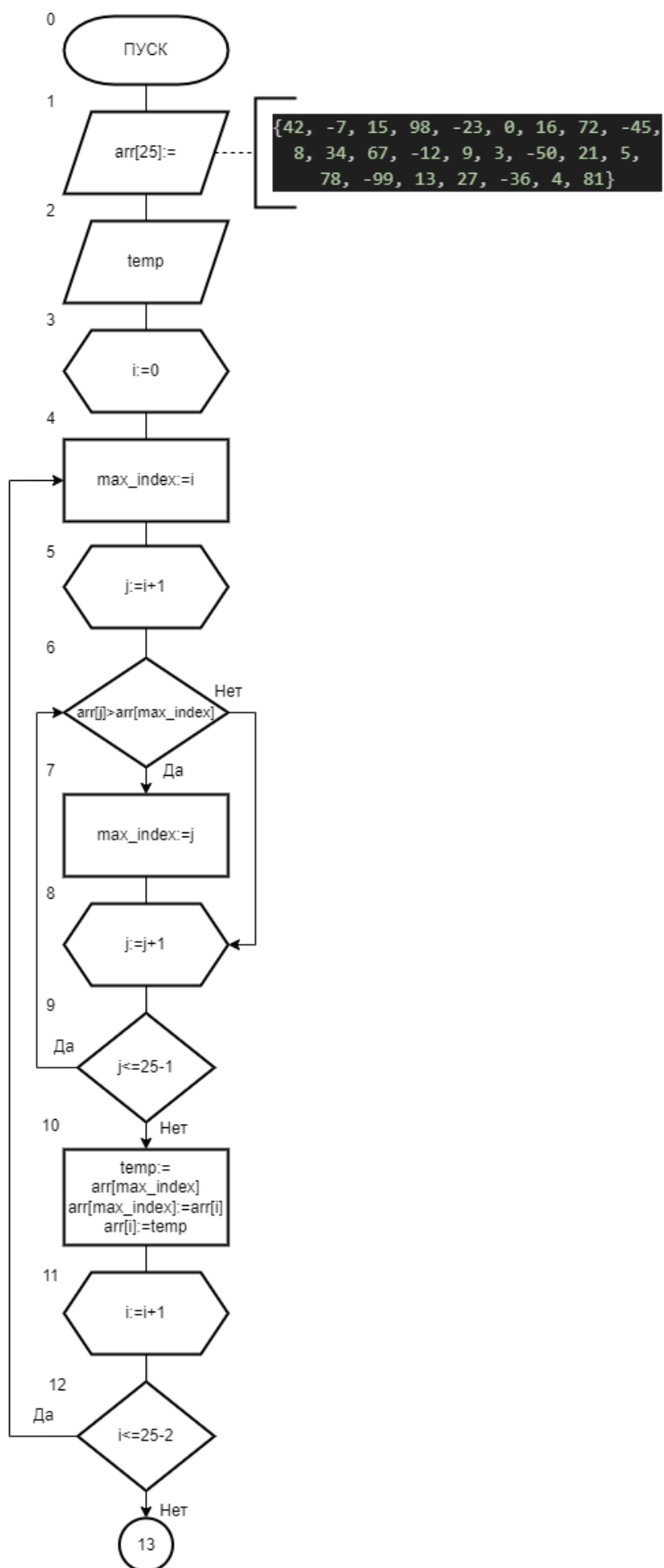
Данный метод малоэффективен для больших массивов. Так же его минус в том, что в такой вариации он проходит по массиву даже после того, как он отсортирован.

Задача 2: отсортировать одномерный массив методом выбора.

Математическая модель:



Блок схема:



Список идентификаторов:

Имя	Тип	Смысл
Arr[]	int	Вводимое число
Max_Index	int	Индекс максимального элемента
I	Int	Параметр цикла
J	int	Параметр цикла

Код программы:

```
#include <stdio.h>
int main(void){

int arr[25] = {42, -7, 15, 98, -23, 0, 16, 72, -45, 8, 34, 67, -12, 9, 3, -50, 21, 5, 78, -99, 13, 27, -36, 4, 81};
int temp;

for (int i = 0; i <= 25 - 2; i++)
{
    int max_index = i;
    for ( int j = i+1; j <= 25 - 1; j++)
    {
        if (arr[j] > arr[max_index]){
            max_index = j;
        }
    }
    temp = arr[max_index];
    arr[max_index] = arr[i];
    arr[i] = temp;
}

for (int i = 0; i <= 25-1; i++)
{
    printf("%d ", arr[i]);
}

}
```

Результат работы программы:

```
98 81 78 72 67 42 34 27 21 16 15 13 9 8 5 4 3 0 -7 -12 -23 -36 -45 -50 -99
```

Анализ результатов вычислений:

Анализируя код, понимаем, что все работает отлично, без ошибок.

Вывод:

средствами языка C я научился работать с итерационными циклическими вычислительными процессами с управлением по индексу и функции и решил поставленные задачи.