

Одномерный массив

Массив – это набор данных, элементы которого образуют упорядоченную последовательность, причем все эти элементы должны быть одного типа.

Как и обычные переменные, массивы в Си описываются в начале функции или в начале блока, причем при описании массива, как правило, указывается количество ячеек, под которое будет выделяться память (статический массив). В Си существует возможность динамического выделения памяти для оптимизации использования ресурсов исполнителя.

Однако в наших лабораторных работах будем использовать статические массивы.

Общая форма описания одномерного статического массива:

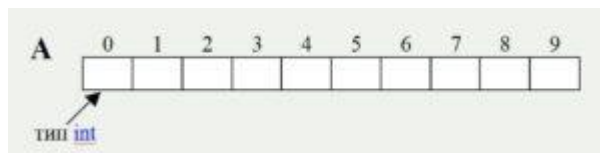
```
тип имя_массива [размер];
```

Пример

Описание массива из 10 целых чисел (тип `int`) с именем `A`.

```
int A[10];
```

В результате, **в памяти компьютера выделяется 10 ячеек типа `int`**. Если одна ячейка занимает 2 байта, то всего будет выделено 20 байт памяти. Эти ячейки объединены общим именем `A`. Нумерация элементов начинается с 0.



Инициализация массива может выполняться в фигурных скобках после знака равно, каждый элемент массива отделяется от предыдущего запятой.

```
int A[] = {5, -12, -12, 9, 10, 0, -9, -12, -1, 23}; // инициализации массива без определения его размера
```

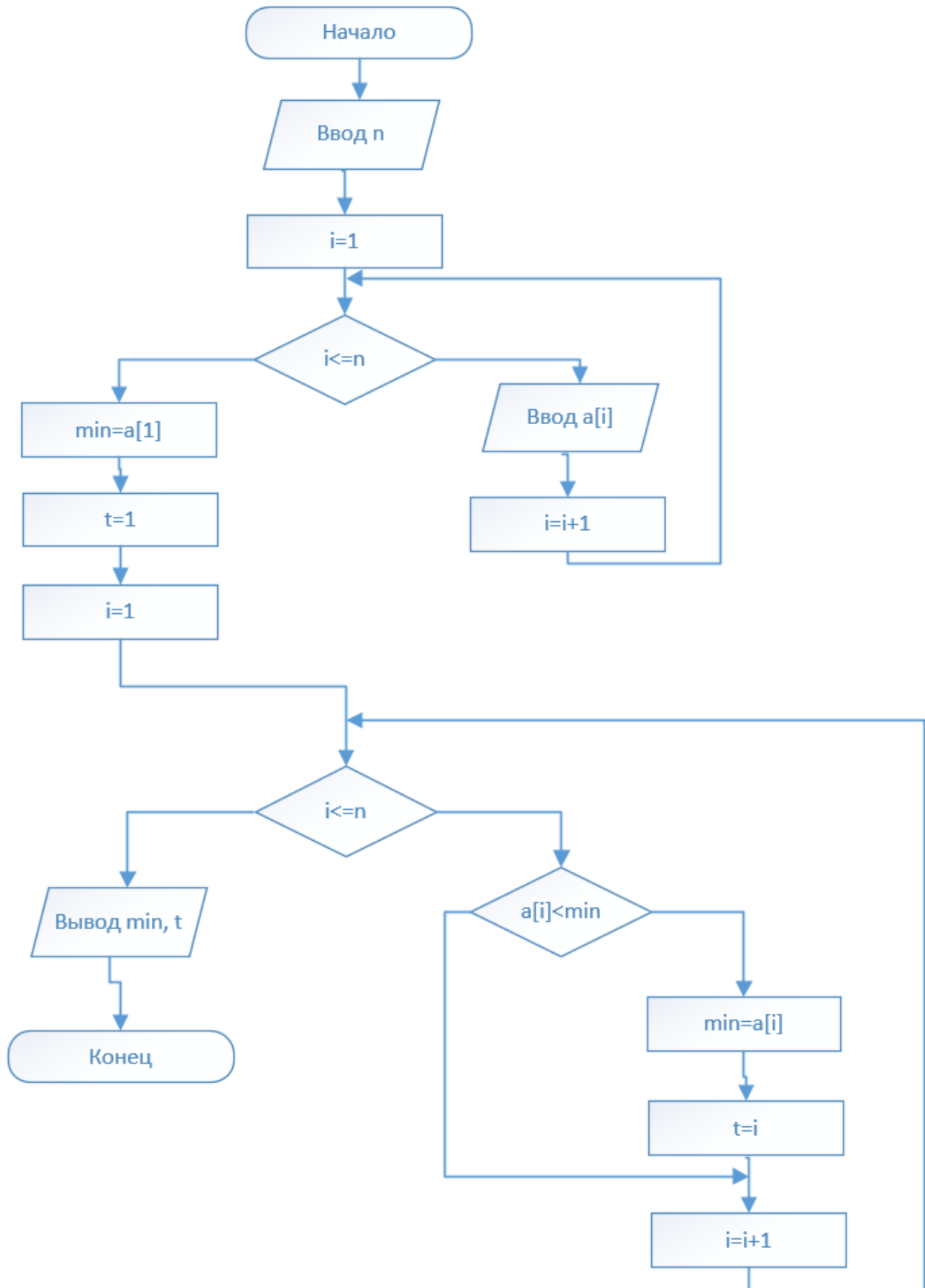
Доступ к отдельному элементу массива может осуществляться с помощью индекса. Индекс определяет позицию элемента в массиве. Первый элемент массива имеет нулевой индекс, последний $n-1$.

`A[0]`, `A[1]`, `A[2]`, `A[3]`, `A[4]`, `A[5]`, `A[6]`, `A[7]`, `A[8]`, `A[9]`.

Есть и другие способы обращения к элементам массива. Они будут подробно рассмотрены, на лекциях.

1. Решение задачи:

Задача: Дан массив действительных чисел длиной n . Найти \min элемент массива и его номер.



2. Пишем программу:

```
#include<stdio.h>
#include<locale.h>
#pragma warning (disable : 4996)
void main(void)
{
    setlocale(LC_ALL, "Russian");
    int n, i, nom;
    float a[100], min;
    printf("Введите размер массива\n");
    scanf("%d", &n);
    printf("Введите элементы \n");
    i = 1;
    while (i <= n) {
        scanf("%f", &a[i]);
        i = i + 1;
    }
    min = a[1];
    nom = 1;
    i = 1;
    while (i <= n) {
        if (a[i] < min) {
            min = a[i];
            nom = i;
        }
        i = i + 1;
    }
    printf("min=%f и его номер %d",min, nom);
}
```

n - количество элементов массива, которые будут использоваться

i - счетчик итераций циклов

nom - переменная под номер минимума

Все эти числа должны быть целыми, поэтому **int**.

Так как массив состоит из действительных чисел, его минимальный элемент тоже будет типа float.

Задаем **i** равным 1, значит будем использовать элементы массива ,с **a[1]** по **a[n]**.

Для заполнения массива будем использовать основную структуру повторения — цикл с предусловием **while**.

while (условие)
инструкция;

В инструкции надо добавить смену счетчика во избежание заикливания.

i <= **n** - условие выхода из цикла.

Пока условие истинно, выполняется инструкция, затем повторяется проверка условия. Когда условие станет ложным, цикл прервется.

В качестве отправной точки при поиске минимума возьмем значение первого элемента массива **a[1]**. Затем в цикле сравним это значение со всеми элементами массива. При выполнении условия **a[i] < min** предыдущее значение **min** меняем на текущее. И запишем номер этого элемента в переменную **nom**.

Лабораторная работа №2.

Для выполнения лабораторной работы необходимо:

1. Получить индивидуальное задание.
2. Составить блок схему алгоритма решения задачи.
3. Написать программу на языке программирования Си в соответствии с алгоритмом, описанным блок схемой.
4. **ОТЛАДИТЬ И ПРОТЕСТИРОВАТЬ ПРОГРАММУ.**
5. Составить отчет о лабораторной работе.

Содержание отчета:

- Титульный лист.
- Условие задачи.
- Блок схема.
- Текст программы.
- Скриншоты, подтверждающие корректную работу программы с разными наборами входных данных (около 5 наборов).