

Программирование на языке C



Модуль 6. **МАССИВЫ**

- Декларация массивов и их размещение в памяти
- Индексация элементов массива
- Массивы переменной длины (VLA)
- Алгоритмы суммирования, поиска и сортировки



Массив - конечная совокупность данных одного типа, имеющая общее имя.

Определяющие свойства массивов:

1. В массиве хранятся отдельные значения, которые называются элементами;
2. Все элементы массива должны быть одного типа. Элементами массива не могут быть функции и объекты типа void;
3. Все элементы массива хранятся в памяти последовательно; первый элемент имеет нулевое смещение адреса, т.е. нулевой индекс;
4. Имя массива является константой, содержит адрес первого элемента массива (является указателем на массив).

Классификация
массивов

Массивы	Статические:	Одномерные	<ul style="list-style-type: none">- их размер изменить нельзя- время жизни определяется по известным правилам
		Многомерные	
	Динамические		<ul style="list-style-type: none">- их размер можно изменять в ходе выполнения программы- время жизни может определяться программистом

Определение массивов.

Обращение к элементам массивов

- **Определение массива как одномерного набора данных одного типа**

<имя типа> <идентификатор> [<размер>];,***

- * — здесь парные квадратные скобки [] являются элементом грамматики, а не метасимволом описания;
- ** — в языке Си имеются средства поддержки только одномерных массивов, однако если входящие в массив элементы также являются одномерными массивами данных, то по индукции может быть определен и *n*-мерный массив (см. ниже)

- **Определение массива как *n*-мерного набора данных одного типа**

*<имя типа> <идентификатор> [<размер 1>][<размер 2>]
<...>[<размер n>];*

- **Обращение к элементу *n*-мерного массива**

*<идентификатор> [<индексное выражение 1>][<индексное
выражение 2>]<...>[<индексное выражение
n>];**

Размер статического массива должен быть известен заранее и не может быть изменен в ходе выполнения программы!

Размещение массива в памяти

Элементы массива с первого до последнего запоминаются в последовательно возрастающих адресах памяти. Многомерные массивы в памяти запоминаются таким образом, что, если переходить от элемента к элементу последовательно, последний индекс изменяется быстрее всего.

Чтобы вычислить количество памяти, занимаемое массивом, используйте оператор **sizeof** (работает только там, где массив определён), возвращающий размер указанного операнда в байтах.

Выход за пределы массива

Необходимо помнить, что при осуществлении доступа к элементам массива нельзя допускать выхода за пределы массива, что, как правило, приводит к сбою программы.

Поэтому:

1. Индекс в индексном выражении не может быть равным или превышать количество элементов в соответствующем измерении массива.
2. Смещение в адресном выражении не может быть равным или превышать количество элементов массива.
3. Индекс ≥ 0 .

Инициализация массивов

- **Полная инициализация одномерного массива без указания его размера** (определяется длиной списка значений)

*<имя типа> <идентификатор> [] = {<список значений>};**

- **Частичная или полная инициализация одномерного массива с указанием его размера**

*<имя типа> <идентификатор> [<размер>] = {<список значений>};**

- * — здесь парные квадратные скобки [] являются элементом грамматики, а не метасимволом описания;

- **Частичная или полная инициализация *n*-мерного массива с указанием его размеров**

<имя типа> <идентификатор> <список размеров> = {<список (не)полных списков значений массива по каждому измерению>};

Массивы переменной длины (VLA)

- **Статический массив**

```
#define N    5  
int data[N] = { [3]=1 };    // 0 0 0 1 0
```

- **VLA массив**

```
int n=5;  
int data[n];    // нельзя инициализировать!
```

Зачем? Плюсы?

Массивы и указатели

- Согласно синтаксису языка Си идентификатор массива без индексов элементов — это указатель-константа со значением адреса нулевого (имеющего индекс [0]) элемента массива



Массивы и функции

Массивы в функцию передаются фактически всегда по адресу, т.е. через указатель (также **требуется** передать количество элементов по каждой из размерностей массива).

Указатели на статические массивы определяются адресом первого элемента массива (элемента с нулевыми индексами).

Детальнее:

Одномерные статические массивы передаются идентификатором (т.е. по имени),
многомерные статические массивы – адресом элемента со всеми нулевыми индексами

Алгоритмы сортировки и поиска

- **Алгоритмы сортировки**

- «пузырьковая» сортировка;
- сортировка вставками;
- сортировка Д. Шелла;
- быстрая сортировка (quick sort)

- **Алгоритмы поиска**

- линейный поиск в неупорядоченном массиве;
- бинарный поиск (дихотомия) в упорядоченном массиве

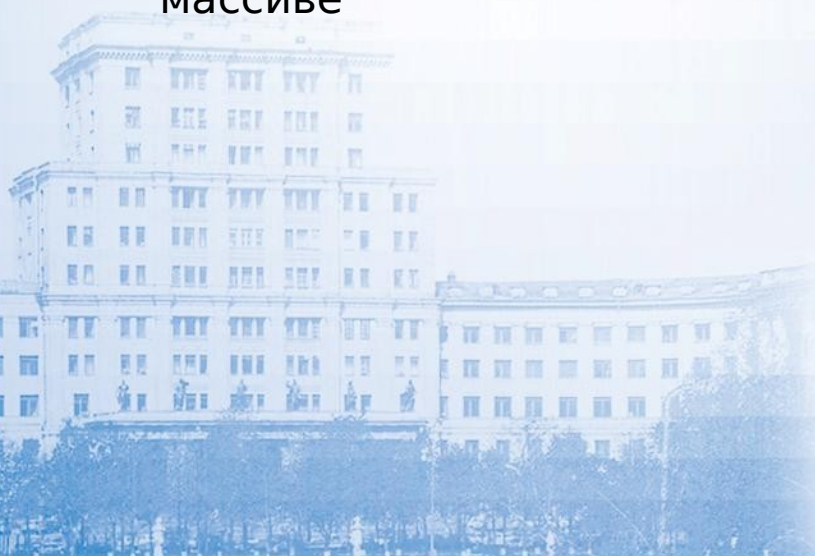
Практика / ДЗ

- **Алгоритмы сортировки**

- Реализовать «пузырьковую» сортировку;
- Реализовать сортировку простой выборкой;
- Реализовать сортировку подсчётом

- **Алгоритмы поиска**

- Реализовать алгоритм бинарного поиска в упорядоченном массиве



Список литературы

- [КР92] Керниган Б., Ритчи Д. Язык программирования Си / Пер. с англ. — М.: Финансы и статистика, 1992. — 272 с.
- [КР06] Керниган Б., Ритчи Д. Язык программирования С / Пер. с англ. — М.: Вильямс, 2006. — 304 с.
- [Под04] Подбельский В.В., Фомин С.С. Программирование на языке Си. – 2-е доп. изд. – М., Финансы и статистика, 2004. – 600 с.
- [Кнут08] Кнут Д.Э. Искусство программирования / Пер. с англ. — Т. 3. Сортировка и поиск. — 2-е изд. — М.: Вильямс, 2008. — 824 с.

