

Массивы и указатели

Что произойдет?

```
1. int a = 10;  
2. int* ptr_a = &a;  
3. *ptr_a++;  
4. cout << a;
```

Адресная арифметика

В примере на предыдущем слайде приоритет у инкремента выше, чем у операции разыменования, поэтому сначала адрес увеличивается на 1.

Но почему это действие не вызывает ошибку компиляции?

Адресная арифметика

При сложении с целым числом, указатель сдвигается на другую ячейку памяти.

Т.е. сложение и вычитание с целыми числами работают с указателями.

Указатели и массивы

Что из себя представляет массив на самом деле? В памяти это непрерывный блок ячеек памяти, а имя массива - указатель на его первый элемент. А последующие элементы получаются как раз сдвигом этого указателя.

Именно поэтому индекс первого элемента массива - 0, потому что никакого сдвига нет

<algorithm>

Разберем некоторые полезные функции, которые могут работать с массивами, но как параметры принимают указатели (а некоторые и как результат также возвращают указатели)

Функции <algorithm>

`sort(указатель на начало, указатель на конец)`

Данная функция сортирует массив в заданном промежутке по неубыванию. Пример:

```
sort(a, a + n); // отсортирует первые n элементов
```

`is_sorted(начало, конец)`

Функция, проверяющая, отсортирован ли заданный диапазон по неубыванию. Как результат возвращает `true` или `false`

Функции <algorithm>

`reverse(указатель на начало, указатель на конец)`

Данная функция переворачивает последовательность в заданном диапазоне:

`reverse(a, a + n);` // перевернет первые n элементов

`count(начало, конец, значение)`

Функция, определяющая количество элементов в заданном диапазоне, равных указанному значению

Функции <algorithm>

`replace(начало, конец, старое значение, новое значение)`

Данная функция заменяет все элементы, равные одному значению на новое значение. Пример:

```
replace(a, a + n, 1, 2); // заменит все единицы на двойки
```

`equal(начало первого диапазона, конец первого диапазона, начало второго диапазона)`

Функция, определяющая, равны ли два заданных диапазона. Для второго диапазона не задается конец, потому что предполагается, что они должны быть одинаковых размеров.

Функции <algorithm>

`min_element(начало, конец)`

Данная функция возвращает указатель на первый минимальный элемент в заданном диапазоне. Чтобы получить само значение, указатель надо разыменовать:

```
int min = *min_element(a, a + n);
```

`max_element(начало, конец)`

Аналогично, только определяет первый максимальный элемент.