

СОСТАВНЫЕ ТИПЫ ДАННЫХ

← основаны на простых (базовых) типах

Массивы (статические)

intNumbers	1	7	-5	10	2	8	0	-3
floatNumbers	2.5	4.6	32.0	-1.0	1.11			
someWord	'a'	'u'	't'	'u'	'm'	'n'		

Объявление массива:

short days[7];

↑

тип значений
каждого элемента

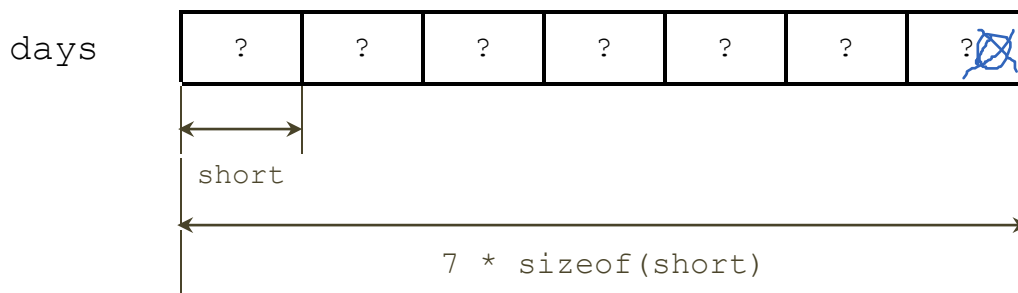
↑

имя массива

↑

количество элементов
в массиве

ИмяТипа имяМассива [размерМассива] ;



Тип значений – любой простой тип, кроме void

Размер массива – должен быть всегда известен:

~~int n;~~
~~cin >> n;~~
~~char days[n];~~

^{int}
const N=10;
 char days[N];

Обращение к элементам:

`int months[12];`

`months[0]` — это первый элемент массива `months`,

`months[11]` — его последний элемент,

`months[20]` — какой элемент?

Пример.

```
int yams[3];
int yamcosts[3] = {20, 30, 5}; // создание и инициализация массива
// Примечание. Если ваш компилятор C++ не может инициализировать
// этот массив, используйте static int yamcosts[3] вместо int yamcosts[3]
cout << sizeof yams; // размер массива в байтах
cout << sizeof yams[0]; // размер одного элемента в байтах
cout << sizeof yams/sizeof (int); // 3 - количество элементов массива
```

Инициализация массивов

```
1. a[0]=45;  
   cin >> a[0];
```

2. Инициализация списком значений (списковая инициализация):

```
①  
int cards [4] = {3, 6, 8, 10};           // все в порядке  
  
②  
int cards [4] = {3, 6, 8, 10, 2, 3}; // ошибка! - значений не должно быть  
                                   // больше указанного кол-ва  
                                   // too many initializers for 'int [4]'
```

```
③  
int hand[4];           // все в порядке  
hand={5, 6, 7, 9}; // списковая инициализация вне объявления переменной  
                  // не допускается  
                  // error: assigning to an array from an initializer list  
  
hand = cards;         // не допускается  
cout << hand;         // error: invalid array assignment
```

```
④  
// меньше значений, чем в массиве объявлено элементов  
float hotelTips[5] = {5.0, 2.5}; 0, 0, 0  
// синтаксис C++11 (без знака =):  
double earnings[4] {1.2e4, 1.6e4, 1.1e4, 1.7e4};
```

```
⑤  
// инициализировать массив нулями  
long totals[500] = {0};  
// синтаксис C++11 (без знака =).  
float balances[100] {}; // Все элементы = 0  
НО:  
float balances[100]; // элементами массива будет «мусор»!
```

```
⑥  
short things [] = {1, 5, 3, 8}; // массив из четырёх элементов  
// размер массива в элементах  
int num_elements = sizeof things/sizeof (short);  
НО:  
// ошибка - размер массива должен быть известен на момент компиляции!  
// error: storage size of 'things' isn't known  
short things [];
```

Пример 1. Заполнение массива данными с клавиатуры

```
const int Num = 5;
int foo [Num];
int n;
cout << "Введите элементы массива:\n";
for ( n=0 ; n<5 ; n=n+1 ) {
    cout << "foo[" << n << "] = ";
    cin >> foo[n];
}
```

Handwritten notes: $n < Num$ and a checkmark next to the input statement.

НО:

```
for ( n=0 ; n<15 ; n=n+1 ) {
    cout << "foo[" << n << "] = ";
    cin >> foo[n];
}
```

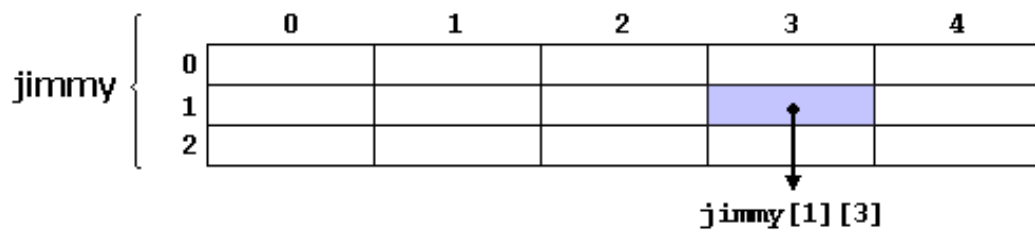
Handwritten notes: A blue 'X' is drawn over the condition $n < 15$, and the variable `foo[n]` is underlined.

Пример 2. Подсчёт суммы элементов массива

```
int foo [] = {16, 2, 77, 40, 12071};
int n, result=0;
for ( n=0 ; n<5 ; n=n+1 )
    result += foo[n];
```

Многомерные массивы (массивы массивов)

```
int jimmy [3][5]; // двухмерный массив
```



Размер массива:

Массив секунды века:

```
char century [100][365][24][60][60];  
требует для хранения своих элементов ??? байт
```

2,975

Инициализация массива:

```
①  
int mas[0][0] = 1; // поэлементная инициализация  
  
②  
int mas[2][3] =  
{  
    {1, 2, 3}, // значения для mas[0]  
    {4, 5, 6}  // значения для mas[1]  
};
```

Пример. Заполнение массива значениями

```
const int WIDTH  = 5,  
        HEIGHT = 3;  
int jimmy [HEIGHT][WIDTH];  
int n,m;  
for (n=0; n<HEIGHT; n++)  
    for (m=0; m<WIDTH; m++)  
    {  
        jimmy[n][m] = (n+1) * (m+1);  
        // или ввод данных пользователем  
        // cin >> jimmy[n][m];  
    }
```

	0	1	2	3	4
0	1	2	3	4	5
1	2	4	6	8	10
2	3	6	9	12	15

Пример. Прочитать значения элементов массива, заданные по столбцам (сначала элементы, начиная с первого, первого столбца, потом второго и т.д.), и вывести их построчно с указанием номера столбца и строки – см. лр №5

Пример. Посчитать сумму элементов закрашенной области массива.

Анализ изменения индексов массива:

row	elem
0	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
1	1, 2, 3, 4, 5, 6, 7, 8
2	2, 3, 4, 5, 6, 7
3	3, 4, 5, 6
4	4, 5
5	4, 5
6	3, 4, 5, 6
7	2, 3, 4, 5, 6, 7
8	1, 2, 3, 4, 5, 6, 7, 8
9	0, 1, 2, 3, 4, 5, 6, 7, 8, 9

	0	elem	→	9
0				
row				
9				

row = 0..9 (или 0..n-1)

elem = row..n-1-row

$r=4 \quad e=5$

```
const int n = 10;
float a[n][n], sum=0;
for (int row=0; row<n; row=row+1)
    for (int elem=row; elem<n-1-row; elem++)
        sum = sum+a[i];
```

Определение нового типа

int B [20][10];

typedef

int TVekt [10]; // T - type
TVekt B[20];

ИЛИ:

```
typedef int TVekt [10];
typedef TVekt TMatrica [20];
TMatrica B;
```

ИЛИ:

```
typedef int TMatrica [10][20];
TMatrica B;
```