

Одномерные статические массивы.

Одномерные статические массивы

Массив — последовательность элементов одного и того же типа, расположенных в памяти друг за другом.

Определение переменной-массива

```
#define N 10  
...  
int a[N];
```

- Тип элемента может быть любым.
- Количество элементов указывается целочисленным константным выражением.
- Количество элементов не может быть изменено в ходе выполнения программы.

Одномерные статические массивы

Использование переменной-массива

```
for (int i = 0; i < N; i++)  
    sum += a[i];
```

- Для доступа к элементу массива используется индекс.
- Индексация выполняется с нуля.
- В качестве индекса может выступать целочисленное выражение (например, $a[i^0/2]$).
- Си не предусматривает никаких проверок на выход за пределы массива.

Операция	Название	Нотация	Класс	Приоритет	Ассоциат.
[. . .]	Индекс	X[Y]	Постфиксная	16	Слева направо

Инициализация переменных-массивов

```
int a[5] = {1, 2, 3, 4, 5};  
// a[0] == 1, a[1] == 2, a[2] == 3, a[3] == 4, a[4] == 5
```

```
int b[5] = {1, 2, 3};  
// b[0] == 1, b[1] == 2, b[2] == 3, b[3] == 0, b[4] == 0
```

```
int c[5] = {0};  
// все элементы массива c - нули
```

```
int d[3] = {1, 2, 3, 4};  
// warning: excess elements in array initializer
```

Инициализация переменных-массивов

```
int f[] = {1, 2, 3, 4, 5};
```

```
// f[0] == 1, f[1] == 2, f[2] == 3, f[3] == 4, f[5] == 5
```

- Компилятор самостоятельно определит количество элементов в массиве `f` и выделит память.
- В программе количество элементов в массиве `f` можно определить из выражения `sizeof(f) / sizeof(f[0]);`

Выделенные инициализаторы (c99 designated initializers)

```
int e[10] = {0, 0, 5, 0, 0, 10, 0, 0, 15, 0};
```

```
// c99
```

```
int e[10] = {[2] = 5, [5] = 10, [8] = 15};
```

Передача массива в функцию

Когда одномерный массив является параметром функции, количество элементов можно не указывать:

```
#define N 10

int sum_array(int a[])
{
    ...
}
```

Как определить какое количество элементов в массиве?

Передача массива в функцию

Передается параметр, который содержит количество элементов в массиве.

```
int sum_array(int a[], int n)
{
    int sum = 0;

    for (int i = 0; i < n; i++)
        sum += a[i];

    return sum;
}
```

У функции нет никакого (!) способа проверить правильность указанного количества элементов массива.

Передача массива в функцию

Функция может изменять значение элементов массива, используемого при вызове.

```
void zero(int a[], int n)
{
    for (int i = 0; i < n; i++)
        a[i] = 0;
}

//...
int a[] = {1, 2, 3, 4, 5};

zero(a, sizeof(a) / sizeof(a[0]));

for (int i = 0; i < sizeof(a) / sizeof(a[0]); i++)
    printf(" %d", a[i]);
// 0 0 0 0 0
```


Передача массива в функцию

Если требуется запретить изменение массива, передаваемого в функцию, используется модификатор `const`.

```
void print(const int a[], int n)
{
    printf("Array: ");
    for (int i = 0; i < n; i++)
        printf(" %d", a[i]);
    printf("\n");

    /*
    a[0] = 0; // error: assignment of read-only location '*a'
    */
}
```