**ЛЕКЦИЯ 2. Массивы**

СОДЕРЖАНИЕ

[1 Объявление массива 1](#_Toc145963023)

[2 Примеры работы с массивами 3](#_Toc145963024)

[3 Советы профессионала 5](#_Toc145963025)

[3.1 Проверяйте код на выход за границу массива 5](#_Toc145963026)

[3.2 Не продолжайте обработку массива после того, как результат получен. 7](#_Toc145963027)

[3.3 Как правильно использовать индексы переменных массива? 8](#_Toc145963028)

**Массив** – это структура данных, состоящая из фиксированного числа элементов, имеющих одинаковое имя, одинаковый тип и расположенных в памяти последовательно.

Операция выбора элемента массива называется **индексацией**; синтаксически она записывается как имя массива, за которым в квадратных скобках следует значение индекса (номера) выбираемого элемента.

Массив – это структура данных с произвольным доступом, то есть время доступа к значению элемента массива не зависит от номера элемента.

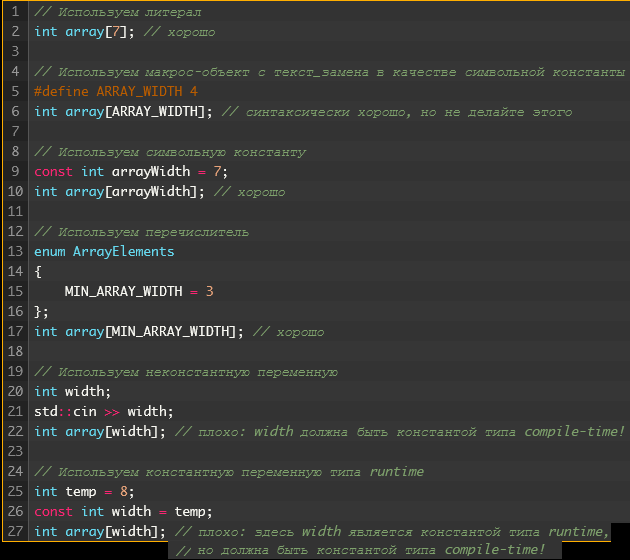
# Объявление массива

Описание массива имеет следующий формат:

**тип имя\_массива[размер\_массива];**

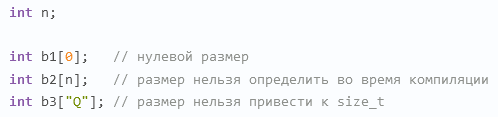
В С++ первый элемент любого массива – это нулевой элемент. Таким образом, индекс первого элемента – 0, второго – 1, третьего – 2 и так далее.

При объявлении массива фиксированного размера, его длина (между квадратными скобками) должна быть **константой типа compile-time** (которая определяется во время компиляции). Вот несколько разных способов объявления массивов с фиксированным размером:



Обратите внимание, в двух последних случаях мы должны получить ошибку, так как длина массива не является константой типа **compile-time**. Некоторые компиляторы могут разрешить использование таких массивов, но они являются некорректными в соответствии со стандартами языка C++ и не должны использоваться в программах, написанных на C++.

Примеры некорректного объявления (<https://habr.com/ru/post/495444/>)



Для объявления массива можно использовать и такой вариант:

**using I4 = int[3];**

**I4 a, b; //Это будет то же самое, что int a[3], b[3];**

**I4 c = { -1, 2, -5 };**

# Примеры работы с массивами

**ПРИМЕР 1**

**#include <iostream>**

**using std::endl;**

**using std::cout;**

**int main() {**

**setlocale(LC\_ALL, "");**

**const int ArraySize=5; // размер массива**

**int a[ArraySize] = {12, 3, 42, 5, 34};// инициализация массива:**

**// a[0]=12, a[1]=3, a[2]=43,**

**// a[3]=5, a[4]=34**

**//начальные значения можно ввести и с клавиатуры:**

**//for (int i=0; i<ArraySize; ++i)**

**// cin>>a[i];**

**for (int i=0; i<ArraySize;++i) //вывод значений элементов**

**cout<<a[i]<<endl; //массива**

**int total=0; //сумма значений элементов массива**

**for (int i=0; i<ArraySize; ++i) //цикл вычисления суммы**

**total+=a[i];**

**cout << "Сумма элементов массива: "<<total<<endl;**

**system("pause");**

**return 0;**

**}**

**ПРИМЕР 2**

**#include <iostream>**

**using namespace std;**

**int main() {**

**int numbers [4]{}; // {0, 0, 0, 0}**

**int numbers1[4]{ 1, 2, 3, 4 };**

**int numbers2[ ]{ 1, 2, 3, 4, 5, 6 }; //размер массива - 6**

**const int numbers3[4]{ 1,2,3,4 };**

**// numbers[1] = 23; // ошибка - значения элементов массива**

**// изменить нельзя**

**int numbers4[]{ 11, 12, 13, 14 };**

**cout << "Length: " << sizeof(numbers4) / sizeof(numbers4[0]) << endl;**

**int count = size(numbers);**

**cout << "Length: " << count << endl; // длина: 4**

**const int max{ 5 }; // максимальная длина массива - не больше 5 чисел**

**int numbers5[max]; // массив для ввода чисел**

**int i{}; // счетчик введенных чисел**

**cout << "Enter numbers" << endl;**

**while (i < max) {**

**cin >> numbers5[i]; // вводим число и изменяем i**

**i++;**

**}**

**// выводим массив**

**for (i = 0; i < max; ++i) {**

**cout << numbers5[i] << "\t";**

**}**

**cout << endl;**

**for (int n : numbers4) {**

**cout << n << " ";**

**}**

**cout << endl;**

**for (auto n : numbers2) {**

**cout << n << endl;**

**}**

**int sum{};**

**for (int i{}; i < size(numbers1); sum += numbers1[i++]);**

**cout << "Sum: " << sum << endl; // Sum: 10**

**}**

# Советы профессионала

**IconAttention**

### Проверяйте код на выход за границу массива

Рассмотрим реализацию алгоритма поиска наименьшего номера K элемента массива A, значение которого равно заданному X:

.

**Вариант 1**

**const int N = 10;**

**int A[N], x, i = 0, k;**

**//Ввод А и х**

**//. . .**

**while ((A[i] != x) && (i < N))**

**i++;**

**if (A[i] == x)**

**k = i;**

**else**

**// {элемент не найден};**

Этот, очевидный на первый взгляд, код содержит фатальную ошибку.

Пусть i = N -1 и A[ i ] не равно X – в этом случае условие продолжения цикла истинно. Тогда i++ равно N, и при вычислении выражения A[ i ] != X в заголовке цикла происходит выход за пределы массива – A[ i ] 🡪 A[N]

**Вариант 2**

Исключим из заголовка цикла проверку A[ i ] != X. Вместо этого будем использовать логическую переменную found.

**const int N = 10;**

**int A[N], x, i = 0, k;**

**bool found = false;**

**//Ввод А и х**

**//. . .**

**while ((!found) && (i < N))**

**if (A[i] == x)**

**found = true; // элемент найден**

**else**

**i++1;**

**if (found)**

**k = i;**

**else**

**//. . . {элемент не найден};**

Добавление логической переменной found сделало код безопасней, но добавило ему сложность: Это, конечно, далеко не лучший вариант поиска!

**Вариант 3 – поиск «с барьером»**

Увеличим размер массива A на 1 и присвоим элементу A[N+1] значение X. A[N+1] – это «барьер».

**const int N = 11;**

**int A[N], x, i = 0, k;**

**A[N - 1] = x;**

**//Ввод А и х**

**//. . .**

**while (A[i] != x)**

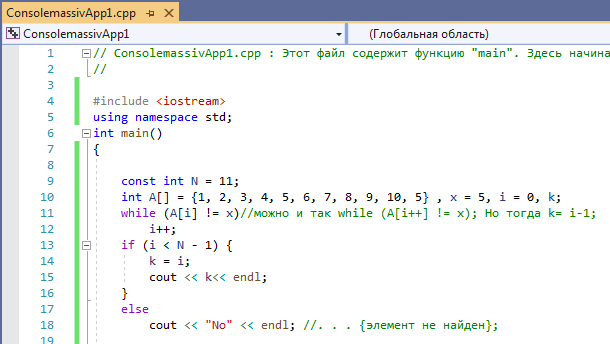
**i++;**

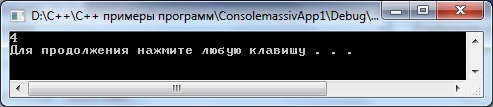
**if (i<N-1)**

**k = i;**

**else**

**//. . . {элемент не найден};**





### Не продолжайте обработку массива после того, как результат получен.

Пусть переменная Found принимает значение true, если   
в массиве есть хотя бы один элемент больше 0, иначе – false.

**Вариант 1**

**const int N = 5;**

**int A[] = {-1, -2, 3, 4, 5};**

**bool found = false;**

**for (int i = 0; i < N; ++i) {**

**if (A[i] > 0)**

**found = true; //элемент найден, но цикл продолжается!**

**}**

**cout << found << endl; //…{1 - элемент найден, 0 - нет**

**Вариант 2**

**const int N = 5;**

**int A[] = {-1, -2, 3, 4, 5};**

**bool found = false;**

**for (int i = 0; i < N; ++i) {**

**if (A[i] > 0) {**

**found = true;**

**break; // элемент найден и выход из цикла**

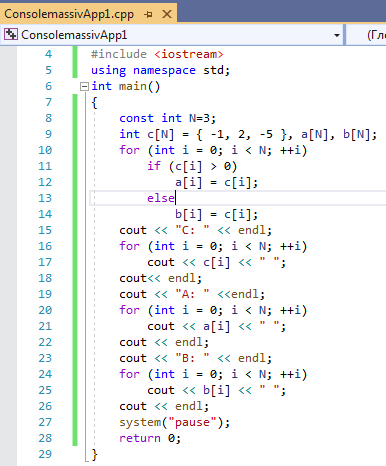
**}**

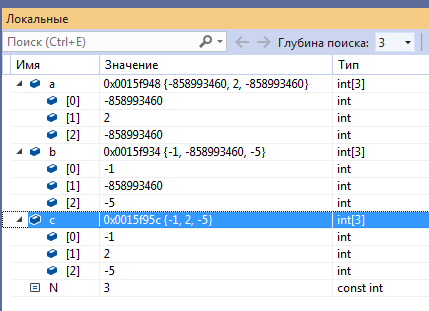
**}**

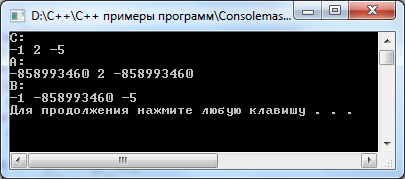
**cout << found << endl; //…{1 - элемент найден, 0 - нет }**

### Как правильно использовать индексы переменных массива?

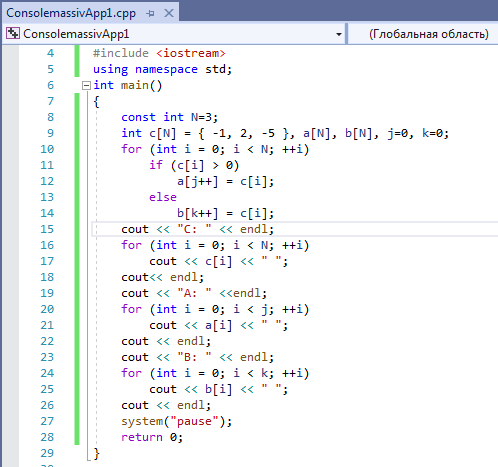
**Задача:** сформировать массив А из положительных элементов массива С, массив В – из отрицательных элементов С.

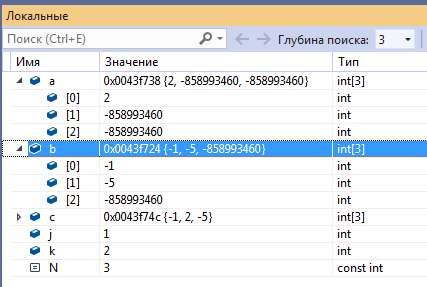


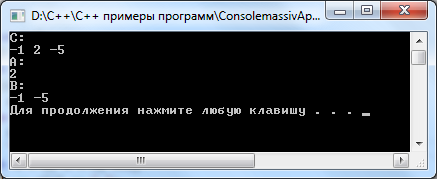




Исправленный вариант







**САМОСТОЯТЕЛЬНО**

В рассмотренной задаче использовать префиксные операции ++ для индексов j, k.

Измениться ли код вывода результатов?