

## Урок 7. Одномерные массивы

В программах очень часто приходится обрабатывать большие объёмы однотипных данных. Применять обычные переменные для этого очень неудобно: представьте себе 1000 переменных с разными именами и одинакового типа, которые нельзя обработать в цикле. Эту проблему позволяет решить использование массивов.

**Массив** – совокупность переменных (элементов) одинакового типа и с общим названием. Доступ к элементам массива осуществляется простым указанием номера элемента – индекса.

*В языке Си нумерация элементов массива начинается с нуля!*

Примеры объявления массивов:

```
int a[10];           // Целочисленный массив a,  
                   // размер – 10 элементов  
double vect[20];   // Массив вещественных чисел vect,  
                   // в нём 20 элементов  
char str[1024];    // Символьный массив на 1024  
                   // элемента
```

При объявлении массива его элементам можно сразу присвоить нужные значения (т.е. инициализировать массив):

```
int x[5] = {10, 20, 30, 40, 50};  
char str[] = "Hello!"; // Автоматическое  
int a[] = {1, 2, 3};   // определение размера
```

Примеры обращения к элементам массива:

```
a = x[0];  
printf("%d\n", x[0]);  
scanf("%d", &x[0]);
```

### Примечания:

1. В памяти компьютера элементы массива располагаются в виде единого блока, друг за другом:

| x[0] | x[1] | x[2] | x[3] | x[4] |
|------|------|------|------|------|
| 10   | 20   | 30   | 40   | 50   |

2. В языке Си нет проверки правильности индекса (т.е. номера элемента). Выход за границы массива чреват непредсказуемыми последствиями.

### **Задание 1**

Заполните одномерный массив размером 10 элементов числами от 1 до 10 и выведите содержимое массива на экран в виде

```
a[0] = 1  
a[1] = 2  
и т.д.
```

### **Решение**

```
#include <stdio.h>  
#include <stdlib.h>  
  
void main()  
{ // Инициализация переменных  
  int a[10]; // Массив a типа int,  
             // размер - 10 элементов  
  
  int i;  
  // Заполнить массив  
  for (i = 0; i < 10; i++)  
    a[i] = i + 1;  
  // Вывести массив на экран  
  for (i = 0; i < 10; i++)  
    printf("a[%d]=%d\n", i, a[i]);  
  // Ожидание нажатия клавиши  
  system("PAUSE");  
}
```

### **Пояснения к тексту программы**

```
int a[10]; // Массив a типа int,  
           // размер - 10 элементов
```

В этой строке происходит объявление массива a. Он имеет целочисленный тип и состоит из 10 элементов, имеющих индексы (номера) 0, 1, 2, 3, 4, 5, 6, 7, 8 и 9.

```
for (i = 0; i < 10; i++)  
  a[i] = i + 1;
```

Цикл заполняет массив a числами от 1 до 10, присваивая значение каждому элементу массива. Переменная i (счётчик цикла) принимает значения от 0 до 9.

```
for (i = 0; i < 10; i++)  
  printf("a[%d]=%d\n", i, a[i]);
```

Цикл выводит массив a на экран. Функция printf применяется для каждого элемента массива. Переменная i (счётчик цикла) «пробегает» все значения от 0 до 9 и элементы массива выводятся один за другим.

## **Задание 2**

Введите одномерный массив из 10 элементов типа `int` с клавиатуры и выведите массив на экран так же, как и в задании 1.

## **Решение**

```
#include <stdio.h>
#include <stdlib.h>

void main()
{ // Инициализация переменных
  int a[10]; // Массив a типа int,
             // размер - 10 элементов
  int i;
  // Заполнить массив
  for (i = 0; i < 10; i++)
  {
    printf("Введите a[%d]: ", i);
    scanf("%d", &a[i]);
  }
  // Вывести массив на экран
  for (i = 0; i < 10; i++)
    printf("a[%d]=%d\n", i, a[i]);
  // Ожидание нажатия клавиши
  system("PAUSE");
}
```

## **Пояснения к тексту программы**

```
// Заполнить массив
for (i = 0; i < 10; i++)
{
  printf("Введите a[%d]: ", i);
  scanf("%d", &a[i]);
}
```

В этом цикле происходит ввод массива `a` с клавиатуры. Каждый элемент массива вводится по отдельности. Переменная `i` (счётчик цикла) – индекс массива, она принимает значения от 0 до 9.

Сначала функция `printf` выводит «подсказку», а затем `scanf` запрашивает значение элемента массива с клавиатуры. `&a[i]` означает адрес элемента `a[i]`.

### **Задание 3**

Инициализируйте одномерный массив в тексте программы и выведите его на экран

### **Решение**

```
#include <stdio.h>
#include <stdlib.h>

void main()
{ // Инициализация переменных
  int a[10] = {10,20,30,40,50, // Массив а типа int,
              9 ,8 ,7, 6, 5}; // размер - 10 элементов

  int i;
  // Вывести массив на экран
  for (i = 0; i < 10; i++)
    printf("a[%d]=%d\n", i, a[i]);
  // Ожидание нажатия клавиши
  system("PAUSE");
}
```

### **Пояснения к тексту программы**

```
int a[10] = {10,20,30,40,50, // Массив а типа int,
            9 ,8 ,7, 6, 5}; // размер - 10 элементов
```

Здесь объявляется массив а типа int размером 10 элементов. В этой же строке он заполняется числами, указанными в фигурных скобках.

```
for (i = 0; i < 10; i++)
  printf("a[%d]=%d\n", i, a[i]);
```

Вывод всех элементов массива на экран. Переменная i (счётчик цикла) «пробегают» все значения от 0 до 9 и элементы массива выводятся один за другим.

#### **Задание 4**

Заполните одномерный массив `int a[10]` случайными числами, скопируйте его в массив `int b[10]`. Выведите оба массива на экран.

#### **Решение**

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

void main()
{ // Инициализация переменных
  int a[10], b[10]; // Массивы a и b типа int,
                  // размер - 10 элементов

  int i;
  // Сброс генератора случайных чисел
  srand(time(NULL));
  // Заполнить массив случайными числами
  for (i = 0; i < 10; i++)
    a[i] = rand();
  // Скопировать массив a в массив b
  for (i = 0; i < 10; i++)
    b[i] = a[i];
  // Вывести массивы на экран
  for (i = 0; i < 10; i++)
    printf("a[%d]=%6d b[%d]=%6d\n", i, a[i], i, b[i]);
  // Ожидание нажатия клавиши
  system("PAUSE");
}
```

#### **Пояснения к тексту программы**

```
srand(time(NULL));
```

Сброс генератора случайных чисел с помощью текущего времени. Если его не сбрасывать, программа будет каждый раз выводить одни и те же случайные числа.

```
for (i = 0; i < 10; i++)
  a[i] = rand();
```

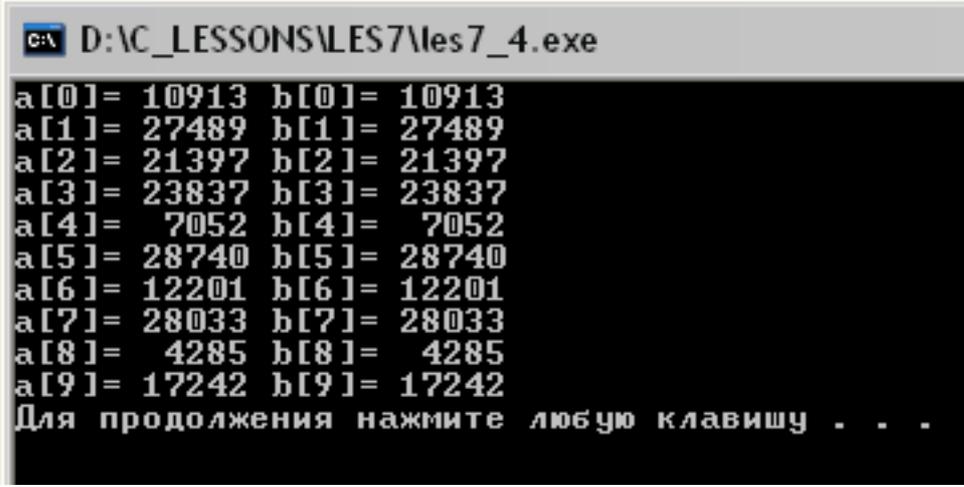
Заполнение массива `a` случайными числами. Переменная `i` принимает значения от 0 до 9. Функция `rand()` вызывается 10 раз (отдельно для каждого элемента).

```
for (i = 0; i < 10; i++)  
    b[i] = a[i];
```

В этом цикле происходит копирование массива *a* в массив *b*. Для этого значение из *a*[0] копируется в *b*[0], из *a*[1] в *b*[1] и т.д. до *a*[9] и *b*[9].

```
for (i = 0; i < 10; i++)  
    printf("a[%d]=%6d b[%d]=%6d\n", i, a[i], i, b[i]);
```

В этом цикле происходит вывод массивов *a* и *b*. Каждый вызов функции `printf` выводит строку со значениями элементов *a*[*i*] и *b*[*i*]. В результате на экране будут показаны оба массива:



```
C:\ D:\C_LESSONS\LES7\les7_4.exe  
a[0]= 10913 b[0]= 10913  
a[1]= 27489 b[1]= 27489  
a[2]= 21397 b[2]= 21397  
a[3]= 23837 b[3]= 23837  
a[4]=  7052 b[4]=  7052  
a[5]= 28740 b[5]= 28740  
a[6]= 12201 b[6]= 12201  
a[7]= 28033 b[7]= 28033  
a[8]=  4285 b[8]=  4285  
a[9]= 17242 b[9]= 17242  
Для продолжения нажмите любую клавишу . . .
```

### **Задание 5**

Найдите максимум массива – элемент с наибольшим численным значением. Используйте следующий алгоритм:

1. Заполните массив `int a[10]` случайными числами
2. В переменную `max` поместите элемент массива `a[0]`, а в переменную `index` – 0.
3. Перейти к следующему элементу массива
4. Если элемент массива больше значения переменной `max`, то поместить его значение в переменную `max`, а его индекс в переменную `index`
5. Если текущий элемент – не последний, то перейти к пункту 2
6. Вывести массив `a` и значение переменных `max` и `index`

### **Решение**

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>

void main()
{
    int a[10];
    int i, max, index;
    // Заполнить массив случайными числами
    srand(time(NULL)); // Сброс генератора случайных чисел
    for (i = 0; i < 10; i++) // Заполнение массива
        a[i] = rand();
    // Найти максимум
    max = a[0]; // В самом начале считаем,
    index = 0; // что a[0] - максимум
    for (i = 1; i < 10; i++) // Поиск по всему массиву
    {
        if (max < a[i])
        {
            max = a[i];
            index = i;
        }
    }
    // Вывести массив на экран
    for (i = 0; i < 10; i++)
        printf("a[%d]=%d\n", i, a[i]);

    // Вывести максимум на экран
    printf("max = %d  index = %d\n", max, index);
    // Ожидание нажатия клавиши
    system("PAUSE");
}
```

### Пояснения к тексту программы

```
max = a[0]; // В самом начале считаем,  
index = 0; // что a[0] – максимум
```

В самом начале считаем, что элемент  $a[0]$  – максимум, т.е. его значение самое большое в массиве  $a$ . В переменной  $max$  хранится значение максимума, а в  $index$  – номер элемента массива, в котором он находится.

```
for (i = 1; i < 10; i++) // Поиск по всему массиву  
{
```

Цикл, в котором происходит поиск максимума по всему массиву. При этом перебираются все элементы, за исключением нулевого (т.е. с номерами от 1 до 9).

```
    if (max < a[i])  
    {  
        max = a[i];  
        index = i;  
    }
```

В теле цикла значение элемент массива  $a[i]$  сравнивается со «старым» максимумом, который хранится в переменной  $max$ . Если  $a[i]$  больше  $max$ ,  $a[i]$  считается «новым» максимумом, а значения переменных  $max$  и  $index$  обновляются.

После того, как цикл «пройдёт» по всему массиву, будет найден настоящий максимум.

```
    }  
    Конец цикла по поиску максимума
```

```
for (i = 0; i < 10; i++)  
    printf("a[%d]=%d\n", i, a[i]);
```

Вывод массива на экран. Это нужно для того, чтобы было можно проверить, правильно ли найден максимум.

```
    printf("max = %d  index = %d\n", max, index);
```

Здесь выводится значение максимума и его индекс (т.е. номер элемента массива, в котором он находится).

### **Задание 6**

Введите массив `double x[10]` с клавиатуры и найдите сумму и среднее арифметическое его элементов

### **Решение**

```
#include <stdio.h>
#include <stdlib.h>

void main()
{
    double x[10];
    double sum, avg;
    int i;

    // Ввести массив с клавиатуры
    for (i = 0; i < 10; i++)
    {
        printf("Введите x[%d]: ", i);
        scanf("%lf", &x[i]);
    }
    // Найти сумму
    sum = 0.0;
    for (i = 0; i < 10; i++)
        sum += x[i];
    // Найти среднее арифметическое
    avg = sum / 10.0;
    // Вывести результат
    printf("sum = %lf\n", sum); // Сумма
    printf("avg = %lf\n", avg); // Среднее арифметическое
    // Ожидание нажатия клавиши
    system("PAUSE");
}
```

### **Пояснения к тексту программы**

```
sum = 0.0;
```

Обнуление переменной, в которой будет храниться сумма всех элементов массива.

```
for (i = 0; i < 10; i++)
    sum += x[i];
```

В этом цикле находится сумма всех элементов массива. Для этого к переменной `sum` по очереди прибавляется каждый элемент массива.

```
avg = sum / 10.0;
```

Для нахождения среднего арифметического сумма элементов массива делится на число элементов массива (в данном случае 10).

## Резюме

**Массив** – совокупность переменных (элементов) одинакового типа и с общим названием. Доступ к **элементам массива** осуществляется простым указанием номера элемента – **индекса**.

***В языке Си нумерация элементов массива начинается с нуля!***

Примеры объявления массивов:

```
int a[10];           // Целочисленный массив a,  
                   // размер - 10 элементов  
double vect[20];    // Массив вещественных чисел vect,  
                   // в нём 20 элементов  
char str[1024];     // Символьный массив на 1024  
                   // элемента
```

При объявлении массива его элементам можно сразу присвоить нужные значения (т.е. инициализировать массив):

```
int x[5] = {10, 20, 30, 40, 50};  
char str[] = "Hello!"; // Автоматическое  
int a[] = {1, 2, 3};    // определение размера
```

Примеры обращения к элементам массива:

```
a = x[0];  
printf("%d\n", x[0]);  
scanf("%d", &x[0]);
```

***В языке Си нет проверки правильности индекса элемента массива. Выход за границы массива чреват непредсказуемыми последствиями.***

## Упражнения

1. Заполните целочисленный массив размером 10 элементов числами от 10 до 1 (т.е.  $a[0] = 10, a[1] = 9, \dots, a[9] = 1$ ). Обязательно используйте цикл `for`.
2. Заполните массив чётными числами ( $a[0]=0, a[1]=2, a[2]=4$  и т.д.).
3. Инициализируйте целочисленный массив в тексте программы (см. задание 3) и выведите квадраты всех элементов массива на экран.
4. Напишите программу по поиску минимума массива (т.е. элемента с наименьшим значением). Используйте текст задания 5.
5. Определить в массиве количество и сумму чётных чисел
6. Определить в массиве количество и сумму нечётных чисел
7. Вывести на экран все элементы массива, равные трём
8. Заполнить массив случайными числами и обнулить все элементы, у которых индекс больше 5.
9. Заполнить массив случайными числами и обнулить все элементы с индексами от 3 до 10
- 10\*. Написать программу, которая находит скалярное произведение векторов по формуле  $ab = a_x b_x + a_y b_y + a_z b_z$ . Для векторов используйте массивы `double a[3]` и `double b[3]`.
- 11\*. Напишите программу кольцевого сдвига элементов массива:

Было:

| a[0] | a[1] | a[2] | a[3] | a[4] |
|------|------|------|------|------|
| 10   | 20   | 30   | 40   | 50   |

Стало:

| a[0] | a[1] | a[2] | a[3] | a[4] |
|------|------|------|------|------|
| 20   | 30   | 40   | 50   | 10   |

12\*\*. В задании 4 урока 6 Вы писали программу, выводящую звёздочки. Доработайте её следующим образом: сначала сразу выводятся 20 разноцветных звёздочек. Затем одна из звёздочек исчезает и появляется в другом, случайном месте; это действие должно повторяться до нажатия клавиши пользователем. Получится что-то похожее на хранитель экрана («скринсейвер») от Norton Commander.

Подсказка: используйте массивы, в которых хранятся координаты  $x$  и  $y$  каждой из звёздочек на экране.