

Лекция 9. Структурированные типы данных

9.1. Массивы

Массив состоит из элементов одного и того же типа. Ко всему массиву целиком можно обращаться по имени. Кроме того, можно выбирать любой элемент массива. Для этого необходимо задать индекс, который указывает на его относительную позицию. Число элементов массива назначается при его определении и в дальнейшем не изменяется. Если массив объявлен, то к любому его элементу можно обратиться следующим образом: указать имя массива и индекс элемента в квадратных скобках. Массивы определяются так же, как и переменные:

```
int a[100]
char b[20]
float d[50]
```

В первой строке объявлен массив **a** из 100 элементов целого типа: **a[0]**, **a[1]**, ..., **a[99]** (индексация всегда начинается с нуля). Во второй строке элементы массива **b** имеют тип **char**, а в третьей - **float**.

Имя массива – это константа, которая содержит адрес его первого элемента (в первом примере, **a** содержит адрес элемента **a[0]**). Предположим, что **a=1000**. Тогда, адрес элемента **a[1]** будет равен 1002 (элемент типа **int** занимает в памяти 2 байта), адрес следующего элемента **a[2]** - 1004 и т.д.

Язык Си позволяет инициализировать массив при его определении. Для этого используется следующая форма:

тип имя_массива[...]={список значений}

Пример 1. Пример программы объявления различных массивов.

```
#include <stdio.h>           //подключение заголовочного файла
#include <stdlib.h>          //для перехода на русский язык

int a[4] = {10,21,32,43};    //объявление массива типа integer
float s[5] = {0.1,0.2,0.3,0.4,0.5}; //объявление массива типа float
char c[3] = { 'a','b','c' }; //объявление массива типа char

void main()                  //основной цикл
{                             //начало основного цикла
    }                         //конец основного цикла
```

Пример 2. Пример программы объявления различных массивов и вывода этих массивов на экран.

```
#include <stdio.h>           //подключение заголовочного файла
#include <stdlib.h>          //для перехода на русский язык

int a[5] = {10,21,32,43,55}; //объявление массива типа integer
```

```

float s[5] = {0.1,0.2,0.3,0.4,0.5};           //объявление массива типа float
char c[5] = { 'a','b','c','d','e' };          //объявление массива типа char
int i = 0;

void main()                                   //основной цикл
{
    system("chcp 1251");                      //начало основного цикла
    system("cls");                            //переходим в консоли на русский язык
    i = 0;                                    //очищаем окно консоли
    while (i < 5)
    {
        printf("Элемент %d - %d - %f - %c\n", i, a[i], s[i], c[i]);
        i++;
    }
    getchar(); getchar();
}                                              //конец основного цикла

```

После выполнения программы на экран будут выведены элементы массивов (рис. 1).

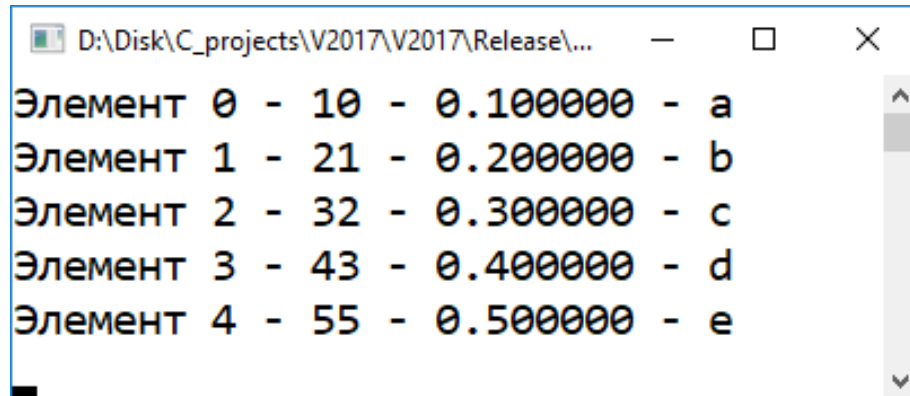


Рис. 1. Программа объявления различных массивов и вывода этих массивов на экран

Пример 3. Пример программы выбора из массива максимального числа и вывода его на экран.

```

#include <stdio.h>                             //подключение заголовочного файла
#include <stdlib.h>                             //для перехода на русский язык

int a[15] = {10,21,32,43,55,-10,5,3,28,91,92,1,2,65,-7};
int i = 0,j;

void main()                                   //основной цикл
{
    system("chcp 1251");                      //начало основного цикла
    system("cls");                            //переходим в консоли на русский язык
    i = 0;                                    //очищаем окно консоли
    j = 0;
    while (i < 15)
    {
        if (j < a[i])

```

```

        {
            j = a[i];
        }
        i++;
    }
    printf("Максимальное число - %d\n", j);
    getchar(); getchar();
}
//конец основного цикла

```

После выполнения программы на экран будут выведено максимальное число массива (рис. 2).

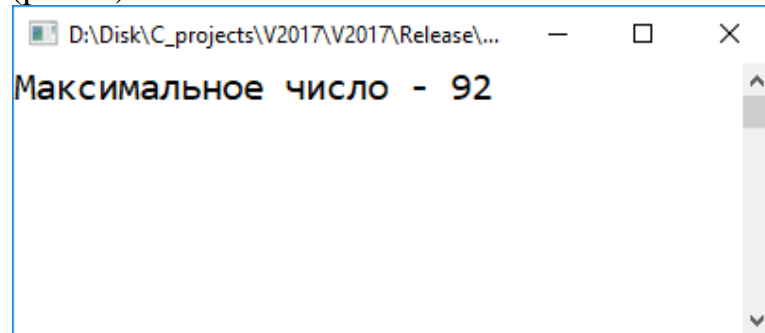


Рис. 2. Программа выбора из массива максимального числа и вывода его на экран

9.2. Структуры

Структура – это объединение одного или нескольких объектов (переменных, массивов, указателей, других структур и т.д.). Как и массив, она представляет собой совокупность данных. Отличием является то, что к ее элементам необходимо обращаться по имени и что различные элементы структуры не обязательно должны принадлежать одному типу.

Объявление структуры осуществляется с помощью ключевого слова **struct**, за которым идет ее тип и далее список элементов, заключенных в фигурные скобки:

```

struct тип { тип элемента_1 имя элемента_1;
            ...
            тип элемента_n имя элемента_n;
        };

```

Именем элемента может быть любой идентификатор. Как и выше, в одной строке можно записывать через запятую несколько идентификаторов одного типа.

```

struct date { int day;
              int month;
              int year;
        };

```

Следом за фигурной скобкой, заканчивающей список элементов, могут записываться переменные данного типа, например:

```

struct date {...} a, b, c;

```

(при этом выделяется соответствующая память). Описание без последующего списка не выделяет никакой памяти; оно просто задает форму структуры.

Введенное имя типа позже можно использовать для объявления структуры, например:

```
struct date days;
```

Теперь переменная *days* имеет тип *date*.

При необходимости структуры можно инициализировать, помещая вслед за описанием список начальных значений элементов.

Разрешается вкладывать структуры друг в друга, например:

```
struct man { char name[20], fam[20];  
            struct date bd;  
            int age;  
        };
```

Определенный выше тип *data* включает три элемента: *day*, *month*, *year*, содержащий целые значения (*int*). Структура *man* включает элементы *name*, *fam*, *bd* и *age*. Первые два - *name*[20] и *fam*[20] - это символьные массивы из 20 элементов каждый. Переменная *bd* представлена составным элементом (вложенной структурой) типа *data*. Элемент *age* содержит значения целого типа *int*. Теперь можно определить переменные, значения которых принадлежат введенному типу:

```
struct man man_[100];
```

Здесь определен массив *man*, состоящий из 100 структур типа *man*.

Чтобы обратиться к отдельному элементу структуры, необходимо указать имя элемента массива, поставить точку и сразу же за ней записать имя нужного элемента структуры, например:

```
man_[j].age = 19;  
man_[j].bd.day = 24;  
man_[j].bd.month = 2  
man_[j].bd.year = 1987;
```