

Основы алгоритмизации и программирования.

Одномерные массивы

Лекция 5

Привезенцев Д.Г.

Муромский институт Владимирского государственного университета
Очная форма обучения

8 октября 2021 г.

Массив: общее определение

Определение

Массив — это непрерывный участок памяти, содержащий последовательность объектов одинакового типа, обозначаемый одним именем.

Массив характеризуется следующими основными понятиями:

Элемент массива (значение элемента массива) — значение, хранящееся в определенной ячейке памяти, расположенной в пределах массива, а также адрес этой ячейки памяти.

Каждый элемент массива характеризуется тремя величинами:

- адресом элемента — адресом начальной ячейки памяти, в которой расположен этот элемент;
- индексом элемента (порядковым номером элемента в массиве);
- значением элемента.

Объявление массива

Массив в Си определяется следующим образом
<тип> <имя массива>[<размер>];

Например,


```
1      int a[100];
```

Получается массив с именем *a*, который содержит сто элементов типа *int*. Как и в случае с переменными, массив содержит мусор.

Для получения доступа до первого элемента, в квадратных скобках пишем его номер (индекс).

Листинг 1: Объявление массива и заполнение некоторых его элементов

```
1 #include <stdio.h>
2 int main()
3 {
4     int a[100];
5     a[0] = 10;
6     a[10] = 333;
7     a[12] = 234;
8     printf("%d %d %d %d", a[0], a[10], a[12], a[99]);
9     getchar();
10 }
```

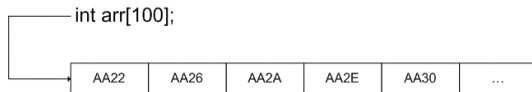


A screenshot of a Windows command prompt window. The title bar reads "C:\WINDOWS\system32\cmd.exe - example1". The window has standard minimize, maximize, and close buttons. The command prompt shows the output of the program: "10 333 234 0". The rest of the window is black, indicating that the program has finished execution and is waiting for input.

Индексация массива

Первый элемент имеет порядковый номер 0.

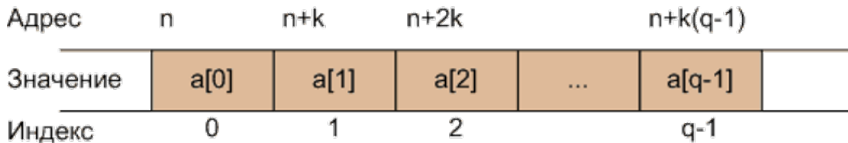
Массив хранит адрес первого элемента. Индекс i элемента - это сдвиг на $i * \text{sizeof}(\text{тип})$ байт от начала.



Индекс массива указывает, на сколько байт необходимо сместиться относительно начала массива, чтобы получить доступ до нужного элемента. Например, если массив A имеет тип `int`, то $A[10]$ означает, что мы сместились на $10 * \text{sizeof}(\text{int})$ байт относительно начала. Первый элемент находится в самом начале и у него смещение $0 * \text{sizeof}(\text{int})$.

Расположение массива в памяти

Графически расположение массива в памяти компьютера можно представить в виде непрерывной ленты адресов.



Представленный на рисунке массив содержит q элементов с индексами от 0 до $q-1$. Каждый элемент занимает в памяти компьютера k байт, причем расположение элементов в памяти последовательное.

Способы объявления массивов

Например, следующее объявление означает, что именем *points* называется массив из 100 действительных чисел.

```
double points[100];
```

В реальных программах следует избегать явного использования числовых констант в объявлениях массива (и других частях программы).

```
double points[100];  
int students[100];
```

```
#define NPOINTS 100  
#define NSTUDENTS 100  
  
double points[NPOINTS];  
int students[NSTUDENTS];
```

Инициализация массивов

Объявление массива может быть совмещено с присвоением значений его элементам. Например:

```
double points[] = {1.0, 3.14, -1.2, 12.65};
```

Инициализация представляет собой набор начальных значений элементов массива, указанных в фигурных скобках, и разделенных запятыми.

```
int a[10] = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9};
```

Если количество инициализирующих значений, указанных в фигурных скобках, меньше, чем количество элементов массива, указанное в квадратных скобках, то все оставшиеся элементы в массиве (для которых не хватило инициализирующих значений) будут равны нулю.

```
int b[10] = {0};
```


Задание начальных значений элементов массива

Листинг 2: Объявление массива и заполнение некоторых его в цикле

```
1  #include <stdio.h>
2  int main()
3  {
4      int a[5]; // объявлен массив a из 5 элементов
5      int i;
6      // Ввод элементов массива
7      for(i = 0; i < 5; i++)
8      {
9          printf("a[%d] = ", i);
10         scanf("%d", &a[i]); // &a[i] - адрес i-го элемента массива
11     }
12     // Вывод элементов массива
13     for (i = 0; i < 5; i++)
14         printf("%d ", a[i]); // пробел в формате печати обязателен
15     getchar();getchar();
16     return 0;
17 }
```

Размер массива

sizeof - это унарный оператор, возвращающий длину в байтах переменной или типа, помещенных в скобки.

Листинг 3: Программное вычисление размера массива

```
1 #include <stdio.h>
2 int main()
3 {
4     int A[57];
5     //sizeof возвращает размер всего массива в байтах
6     //Для определения количества элементов необходимо
7     //разделить размер массива на размер его элемента
8     int size = sizeof A / sizeof(int);
9     printf("Size of array equals to %d", size);
10    getchar();
11    return 0;
12 }
```

Работа с элементами массива

Листинг 4: Заполнение массива

```
1 #include <stdio.h>
2 #define NPOINTS 100
3 int main() {
4     double points[NPOINTS];
5     int k;
6     points[0] = 0.1;
7     for(k = 1; k < NPOINTS; k++)
8     {
9         points[k] = 0.1 + points[k-1];
10    }
11    for(k = 1; k < NPOINTS; k++)
12    {
13        printf("%.1lf ", points[k]);
14    }
15    return 0;
16 }
```

Работа с элементами массива

Листинг 5: Получение размера массива

```
1  #include<stdio.h>
2  int main()
3  {
4      double points[] = {1.0, 3.14, -1.2, 12.65, 56, 76, 87.9};
5      int k;
6      int npoints = sizeof(points) / sizeof(points[0]);
7
8      for(k = 0; k < npoints; k++)
9      {
10         printf("points[%d] = %.2lf\n", k, points[k]);
11     }
12     return 0;
13 }
```

Работа с элементами массива

Листинг 6: Поиск максимума

```
1  #include <stdio.h>
2  #define N 10
3  int main()
4  {
5      int a[N] = {1, 2, 5, 3, 9, 1, 7, 7, 2, 4};
6      int max = 0;
7      for (unsigned i = 1; i < N; i++)
8      {
9          if (a[i] <= a[max])
10         {
11             max = i;
12         }
13     }
14     printf("Min element is a[%d]=%d", max, a[max]);
15     getchar();
16     return 0;
17 }
```

Работа с элементами массива

Листинг 7: Переворачивание массива

```
1 #include <stdio.h>
2 #define SIZE 10u
3 int main()
4 {
5     int A[SIZE] = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9};
6     unsigned half = SIZE / 2; //середина массива
7     //Один счётчик идёт слева направо, другой справа налево
8     for (unsigned i = 0, j = SIZE - 1; i < half; i++, j--) {
9         unsigned tmp = A[i];
10        A[i] = A[j];
11        A[j] = tmp;
12    }
13    for (unsigned i = 0; i < SIZE; i++) {
14        printf("%d ", A[i]);
15    }
16    getchar();
17    return 0;
18 }
```

Работа с элементами массива

Листинг 8: Удаление выбранного пользователем элемента массива

```
1 #include <stdio.h>
2 #define SIZE 10
3 int main()
4 {
5     int A[SIZE] = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9};
6     int index; //индекс, введённый пользователем
7     for (unsigned i = 0; i < SIZE; i++){ //Выводим массив
8         printf("a[%d]=%d ", i, A[i]);
9     }
10    while (1){ //Просим пользователя ввести валидный индекс
11        printf("\nEnter index of element to delete ");
12        scanf("%d", &index);
13        if (index > 0 && index < SIZE)
14            {
15                break;
16            }
17    }
18    //Копируем следующий элемент массива на место удаляемого и так до конца
19    for (unsigned i = index; i < SIZE-1; i++) {
20        A[i] = A[i+1];
21    }
22    for (unsigned i = 0; i < SIZE-1; i++) { //Выводим результат
23        printf("a[%d]=%d ", i, A[i]);
24    }
25    getchar();
26    return 0;
27 }
```

Листинг 9: Ввод различных чисел в массив

```
1 #include <stdio.h>
2 #define SIZE 10u
3 int main() {
4     int A[SIZE] = {0};
5     unsigned i, j;
6     int counter = 1; //сколько разных чисел введено. Как минимум одно.
7     int input;
8     int wasntFound; //флаг, что введённое число не было найдено
9     printf("0. "); //Вводим первое число. Оно ещё не встречалось.
10    scanf("%d", &A[0]);
11    for (i = 1; i < SIZE; i++) {
12        printf("%d. ", i);
13        scanf("%d", &input);
14        wasntFound = 1;
15        //Проверяем, встречалось ли такое число. Если да,
16        //то выставляем флаг и выходим из цикла
17        for (j = 0; j <= counter; j++)
18            if (input == A[j]) {
19                wasntFound = 0;
20                break;
21            }
22        if (wasntFound) { //Если флаг был поднят, то заносим число в массив
23            A[counter] = input;
24            counter++;
25        }
26    }
27    for (i = 0; i < counter; i++)
28        printf("%d ", A[i]);
29    getchar();
30    return 0;
31 }
```


Листинг 10: Заполнение массива разными числами

```
1 #include <stdio.h>
2 #define SIZE 10u
3 int main() {
4     int A[SIZE] = {0};
5     int counter = 0; //сколько разных чисел введено. Как минимум одно.
6     int input;
7     int wasntFound; //флаг, что введённое число не было найдено
8     do {
9         printf("%d. ", counter);
10        scanf("%d", &input);
11        wasntFound = 1;
12        //Проверяем, встречалось ли такое число. Если да,
13        //то выставляем флаг и выходим из цикла
14        for (unsigned j = 0; j < counter; j++)
15            if (input == A[j]) {
16                wasntFound = 0;
17                break;
18            }
19        if (wasntFound) { //Если флаг был поднят, то заносим число в массив
20            A[counter] = input;
21            counter++;
22        }
23        else
24        {
25            printf("Lets input again\n");
26        }
27    } while (counter < SIZE);
28    for (unsigned j = 0; j < counter; j++)
29        printf("%d ", A[j]);
30    getchar();
31    return 0;
32 }
```

Работа с элементами массива

Листинг 11: Случайное перемешивание массива

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <time.h>
4  int main() {
5      const int SIZE = 20; //Используем константу для задания размера массива
6      int A[20] = { 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
7      int i, rnd;
8      int tmp;
9      //Инициализируем генератор псевдослучайных чисел
10     //В качестве начального числа берём системное время
11     srand(time(NULL));
12     for (i = SIZE - 1; i > 0; i--) {
13         rnd = rand() % i; //Случайное число в пределе от 0 до i
14         tmp = A[i];
15         A[i] = A[rnd];
16         A[rnd] = tmp;
17         printf("A[%d]<-->A[%d]\n", rnd, i);
18     }
19     for (i = 0; i < SIZE; i++) {
20         printf("%d ", A[i]);
21     }
22     getchar();
23     return 0;
24 }
```

Задачи

- ❶ Ввести массив, состоящий из 20 элементов целого типа. Определить каких элементов больше четных или нечетных по значению
- ❷ Ввести целочисленный массив, состоящий из 15 элементов. Определить сумму и разность максимального и минимального элементов
- ❸ Ввести массив, состоящий из 16 элементов целого типа. Заменить все элементы, кратные трем, нулями. Определить количество замен
- ❹ В заданном целочисленном массиве $R[9]$ определить индекс наибольшего из нечетных по значению положительных элементов