

Введение в программирование

Лекция 5.

МАССИВЫ



Массив - упорядоченная последовательность пронумерованных элементов одинакового типа.

Индекс - номер элемента массива. Индексы элементов массива начинаются с нуля.

Элемент массива может иметь несколько индексов.

Размерность массива - количество индексов каждого элемента.

Вектор - одномерный массив (один индекс).

Матрица – двумерный массив (первый индекс - номер строки, второй - номер столбца).

В языке С формально имеются только одномерные массивы, а многомерный массив рассматривается как массив массивов.



Объявление массива

```
float x[50];          /* вещественный вектор x */
char st[80];         /* символьный вектор st */
int w[5][10];       /* целочисленная матрица w */
                    /* из 5 строк и 10 столбцов, */
                    /* всего 5*10 = 50 элементов. */
```

Элемент массива - переменная.

Тип элемента указан в объявлении массива.

Обозначение элемента (в языке C) - имя массива и в квадратных скобках индексы.

$x[0]$, $x[1]$, ... , $x[49]$ – элементы массива x ,
вещественные переменные;

$st[0]$, $st[1]$, ... , $st[79]$ – символьные переменные,
 st - текст из 80 символов.



- Элементы матрицы размещаются в памяти строка за строкой в следующем порядке:

$w[0][0], w[0][1], \dots, w[0][9],$

$w[1][0], w[1][1], \dots, w[1][9],$

\dots

$w[4][0], w[4][1], \dots, w[4][9].$

Элементу массива можно присваивать значение, использовать в качестве операнда выражения и т. д.

Основное свойство массива:

все элементы одновременно присутствуют в оперативной памяти, одинаково доступны и могут обрабатываться в любом порядке.



В объявлении массива можно задавать начальные значения его элементов.

- **float** `z[10] = { 2.5, 2, 3.5 };`
`/* z[0]=2.5, z[1]=2, z[2]=3.5, остальные элементы =0 */`
- **int** `m[] = { 2, 3, 2 };` `/* массив из 3-х элементов */`
`/* Количество элементов массива = количеству значений */`
- **char** `t[] = "KGTU";` `/* строка из 5-и символов */`
`/* Эквивалентно 'K', 'G', 'T', 'U', '\0' */`
- **int** `k[2][2] = { {10, 10}, {25, 25}, {5, 5}, {15, 15} };`
`/* Эквивалентно {10, 10, 25, 25, 5, 5, 15, 15} */`



Элементы числового массива нужно
ВВОДИТЬ И ВЫВОДИТЬ ПОЭЛЕМЕНТНО в цикле.

```
#define N 20
float x[N];
int n, i;
scanf("%d",&n);
for(i=0; i<n; i++)
    scanf("%f", &x[i]);
```



Для ввода и вывода символьного массива (строки символов) есть **специальные функции gets() и puts()**.

Функция **gets()** вводит с клавиатуры строку, заменяя символ «перевод строки» нулевым байтом (нуль-символом – ‘\0’).

Функция **puts()** выводит на экран строку, завершающуюся нуль-символом.

```
char st[80];
```

```
gets(st); puts(st);
```



Массив нужен для ВХОДНЫХ или ВЫХОДНЫХ данных, если:

- элементы обрабатываются **неоднократно** или
- не в том порядке, в котором вводятся или выводятся.

В этих случаях приходится одновременно хранить в памяти все элементы в виде массива.

Рассмотрим задачи:

5.1. Дано целое $n > 0$ и последовательность вещественных чисел x_1, x_2, \dots, x_n . Написать программу

- для определения **суммы и количества** положительных элементов последовательности;
- для определения количества элементов последовательности **больших первого элемента** x_1 ;
- для определения количества элементов последовательности **больших последнего элемента** x_n .



Массивы

5.2. Дано целое $n > 0$ и последовательность вещественных чисел x_1, x_2, \dots, x_n . Написать программу

- а) для выдачи на экран **только положительных чисел** последовательности.
- б) для выдачи на экран **сначала всех отрицательных, а затем всех положительных** чисел последовательности.

5.3. Даны значения температур каждого из n дней месяца ($n \leq 31$).
Написать программу

- а) определения **среднемесячной температуры**;
- б) определения **количества дней, когда температура была выше среднемесячной**.

5.4. Дан текст, продолжающийся до конца файла.
Написать программы для решения следующих задач.

- а) Переставить **символы текста в обратном порядке**.
- б) Подсчитать **количество слов**, содержащих букву 'а'.
- в) **Вывести слова**, содержащие букву 'а'.



Массивы

- Задача 5.1. Дано целое $n > 0$ и вещественные x_1, x_2, \dots, x_n . Составить программу подсчета количества чисел больших первого числа - 5.1б (больших последнего числа – 5.1в).
- Тест 5.1б. Вход: 4 3 12 7 5 Тест 5.1в. Вход: 4 3 12 7
5 Выход: k = 3 Выход: k = 2

```
/* Программа 5.1б. */
#include <stdio.h>

void main ()
{ int n, i, k=0;
  float m1, m;
  scanf ("%d", &n);
  scanf ("%f", &m1);
  for (i = 1; i < n; i++)
    { scanf ("%f", &m);
      if (m > m1) k++;
    }
  printf ("k=%d\n", k);
}
```

```
/* Программа 5.1в. */
#include <stdio.h>
#define N 20
void main ()
{ int n, i, k=0;
  float m[N];
  scanf ("%d", &n);
  for (i = 0; i < n; i++)
    scanf ("%f", &m[i]);
  for (i=0; i < n-1; i++)
    if (m[i] > m[n-1]) k++;
  printf ("k=%d\n", k);
}
```



Задача 5.3. Даны значения температур каждого из n дней месяца ($n \leq 31$). Определить среднюю температуру за месяц. Вывести дни, когда температура была выше средней, и количество таких дней.

```
/* Программа 5.3.      Средняя температура      */
#include <stdio.h>
#define N  31
main ()
{ int n;
  int i, k=0;
  float t[N], s=0;
  scanf ("%d", &n);
  for (i = 0; i < n; i++)
  { scanf ("%f", &t[i]);
    s = s + t[i];
  }
```



Массивы

```
s = s/n;
    printf ("Средняя температура =%.1f\n", s);
printf ("\n Дни с температурой выше средней:\n");
for (i = 0; i < n; i++)
    if (t[i] > s)
        { printf ("%d ", i+1);
          k++;
        }
printf ("\n Количество дней = %d", k);
getch();
return 0;
}
```



Массивы

- Задача 5.4а. Переставить символы текста в обратном порядке.
- Тест. Вход: ПИРАМИДА Выход: АДИМАРИП

0 1 2 3 4 5 6 7 - индексы символов в массиве

```
char t[100], s;  int i, j, n;
  gets (t);      /* ввод строки */
n= strlen(t);   /* определение длины строки */
for (i=0, j=n-1; i<j;  i++, j--)
{ s=t[i]; t[i]=t[j]; t[j]=s; }  /* Обмен t[i] <--> t[j] */
```

Для приведенного теста цикл повторяется 4 раза до середины строки, с изменением i от 0 до 3, а j - от 7 до 4.

