

# Правила ввода и решения задач с одномерными массивами. Часть 1

- **массив**
- **описание массива**
- **заполнение массива**
- **ВЫВОД массива**
- **обработка массива**

1. Арифметические функции:

а) для целых чисел: функция  $\text{div}(x)$  определяет целую часть при делении нацело. Например,  $15 \text{ div } 6 = 2$ ; Это деление с остатком:  $15 : 6 = 2*6 + 3$ .  $3$  – это остаток и он отбрасывается.

$20 \text{ div } 3 = 6$ ;  $20 : 3 = 3*3 + 2$  Остаток  $2$  отбрасывается.

$18 \text{ div } 5 = 3$ .  $18 : 5 = 3*5 + 3$  Остаток  $3$  отбрасывается.

Функция  $\text{mod}(x)$  определяет остаток при делении нацело. Например,  $15 \text{ mod } 6 = 3$ ; В этом случае, наоборот, записывается остаток.

$20 \text{ mod } 3 = 2$ ;  $18 \text{ mod } 5 = 3$ .

Решите самостоятельно:

1)  $29 \text{ div } 5 =$        $29 \text{ mod } 5 =$

2)  $25 \text{ div } 7 =$        $25 \text{ mod } 7 =$

3)  $36 \text{ div } 9 =$        $36 \text{ mod } 9 =$



# Массив

**Массив** - это поименованная совокупность однотипных элементов, упорядоченных по индексам, определяющим положение элемента в массиве.

## Одномерный массив



Решение разнообразных задач, связанных с обработкой массивов, базируется на решении таких типовых задач, как:

- суммирование элементов массива;
- поиск элемента с заданными свойствами;
- сортировка массива.

Рассмотрим программы ввода одномерного массива:

С клавиатуры:

Program a1;           - название программы a1  
Const n=10;           - размерность массива  
Var                   - начало раздела описаний  
A:array [1..n] of integer;   - описываем массив A, состоящий из 10 целых чисел  
I:integer;            - параметр цикла  
    Begin               - начало программы  
For I:=1 to n do       - с 1 по 10-ую ячейки памяти заполняем целыми числами,  
Readln(a[I]);         - вводимыми с клавиатуры  
End.                   - конец программы.

С помощью генератора случайных чисел (ввод массива):

Program a1;           - название программы a1  
Const n=10;           - размерность массива  
Var                   - начало раздела описаний  
A:array [1..n] of integer;   - описываем массив A, состоящий из 10 целых чисел  
I:integer;            - параметр цикла  
    Begin               - начало программы  
Randomize;            - подключаем генератор случайных чисел  
For I:=1 to n do       - перемещаясь последовательно от ячейки к ячейке  
  Begin                открываем операторную скобку  
a[I]:=random(10);       - с 1 по 10-ую ячейки памяти заполняем случайными целыми числами,  
write(' ',a[I]:1)       - и выводим их на экран  
  end                   - закрываем операторную скобку  
End.                   - конец программы.

# Вычисление суммы элементов массива

Суммирование элементов массива осуществляется за счёт поочерёдного добавления слагаемых:

→ Определяется ячейка памяти (переменная  $s$ ), в которой будет последовательно накапливаться результат суммирования

→ Переменной  $s$  присваивается начальное значение  $0$  - число, не влияющее на результат сложения

→ Для каждого элемента массива из переменной  $s$  считывается её текущее значение и складывается со значением элемента массива; полученный результат присваивается переменной  $s$ .

# Вычисление суммы элементов массива

$s := 0$	$s = 0$
$s := s + a[1]$	$s = 0 + a[1]$
$s := s + a[2]$	$s = 0 + a[1] + a[2]$
$s := s + a[3]$	$s = 0 + a[1] + a[2] + a[3]$
...	...
$s := s + a[n]$	$s = 0 + a[1] + a[2] + a[3] + \dots + a[n]$

Основной фрагмент программы:

**s:=0;**

**for i:=1 to n do s:=s+a[i];**

# Вычисление суммы элементов массива

```
program n_3;  
  var s, i: integer;  
    a: array[1..10] of integer;  
begin  
  s:=0;  
  randomize;  
  for i:=1 to 10 do a[i]:=random(100);  
  for i:=1 to 10 do write (a[i], ` `);  
  for i:=1 to 10 do s:=s+a[i];  
  writeln ('s=', s)  
end.
```



# Вычисление суммы элементов массива

```
program n_3_1;
  var s, i: integer;
      a: array[1..10] of integer;
begin
  s:=0;
  randomize;
  for i:=1 to 10 do
    begin
      a[i]:=random(100);
      write (a[i], ' ');
      s:=s+a[i]
    end;
  writeln ('s=', s)
end.
```



**Второй способ оформления:**  
Используем операторные скобки и всё (ввод, вывод, обработку) делаем в одном цикле.