

```

3. Basic
INPUT X
L = 0
M = 0
WHILE X > 0
  M = M + 1
  IF X MOD 3 <> 0 THEN
    L = L + 1
  END IF
  X = X \ 3
WEND
PRINT L
PRINT M

```

Для значения 5637 программа выводит 4 и 8.

```

4. C++
int F(int x)
{
  return x*x + 16*x + 15;
}
int main()
{
  int a, b;
  cin >> a >> b;
  int M = 0;
  for (int t = a; t <= b; t++)
    if (F(t) > 0)
      M = M + 1;
  cout << M;
  return 0;
}

```

При  $a = -3$ ,  $b = 5$  программа выводит 6.

3\* Изобразите любой алгоритм из упражнения 2 в виде блок-схемы.



## Глава 1 АЛГОРИТМЫ ОБРАБОТКИ МАССИВОВ

### § 3. Структурированный тип данных массив

Впервые тип данных *массив* появился в языке Фортран (создан в период с 1954 по 1957 г. в корпорации IBM). Уже первые версии языка поддерживали трехмерные массивы (в 1980 г. максимальная размерность массива была увеличена до 7). Массивы были необходимы для создания математических библиотек, в частности содержащих процедуры решения систем линейных уравнений.

**Пример 3.1.** В 10 А классе 25 учащихся. Известен рост каждого в сантиметрах. Для хранения значений роста можно использовать массив  $A$ , состоящий из 25 целых чисел. Индекс каждого элемента — порядковый номер учащегося из списка в классном журнале. Тогда запись  $A[5]$  — рост учащегося под пятым номером.

#### 3.1. Понятие массива

В современном мире каждую секунду происходит обработка огромного числа данных с помощью компьютера. Если необходимо обрабатывать данные одного типа (числа, символы, строки и др.), то для их хранения можно воспользоваться типом данных, который называется **массив**.

**Массив** — упорядоченная последовательность данных, состоящая из конечного числа элементов, имеющих один и тот же тип, и обозначаемая одним именем.

Массив является структурированным (составным) типом данных. Это означает, что величина, описанная

как массив состоит из конечного числа других величин. Так, например, можно создать массивы из 10 целых или 100 вещественных чисел. Тип элементов массива называют **базовым типом**. Все элементы массива упорядочены по индексам, определяющим местоположение элемента в массиве.

Массиву присваивается имя, посредством указания которого можно сослаться на данный массив как на единое целое. Элементы, образующие массив, упорядочены так, что каждому из них соответствует номер (индекс), определяющий место элемента в общей последовательности (примеры 3.1—3.3). Индексы представляют собой выражения любого простого типа, кроме вещественного. Доступ к каждому отдельному элементу осуществляется обращением к имени массива с указанием индекса нужного элемента, индекс элемента записывается после имени в квадратных скобках (пример 3.4).

Если обращение к элементам массива осуществляется при помощи только одного индекса, то такой массив называют **одномерным** или **линейным**. Элементы данного массива располагаются цепочкой друг за другом. Количество индексов, по которым обращаются к элементу в массиве, определяет **размерность массива**. Кроме одномерных, могут использоваться двумерные, трехмерные и другие массивы.

### 3.2. Описание массивов

Описание массива в языке Паскаль происходит следующим образом:

```
var <имя массива>: array [<тип индекса>] of <тип элементов>;
```

**Пример 3.2.** В 10 Б классе 27 учащихся. В классном журнале указаны фамилия и имя каждого из них. Для хранения списка учащихся можно использовать массив S, состоящий из 27 строк. Индекс каждого элемента — порядковый номер учащегося из списка в классном журнале. Тогда запись S[5] — фамилия и имя учащегося под номером 5.

**Пример 3.3.** Каждый день в декабре измеряли температуру воздуха. Для хранения значений температуры можно использовать массив T, состоящий из 31 вещественного числа. Индекс элемента — номер дня в декабре. Запись T[15] — температура воздуха 15 декабря.

**Пример 3.4.** Обращение к элементу массива: a[3], T[i], S[n-1].

Двумерный массив — массив, элементами которого являются одномерные массивы. Его можно представить как таблицу с данными, в которой каждая строка — линейный массив. Обращение к элементу осуществляется по двум индексам: a[3][5] — элемент, находящийся в третьей строке и пятом столбце. Примером использования двумерного массива является лист электронной таблицы.

Массив p, описанный следующим образом:

```
var p: array [1..30] of array [1..30] of boolean;
```

можно использовать для определения свободных мест в зрительном зале. Тогда запись

```
p[2][13] := true;
```

будет означать, что во втором ряду место 13 свободно, а запись

```
p[5][7] := false; —
```

в пятом ряду место 7 занято.

**Пример 3.5.** Опишем массив, рассмотренный в примере 3.1. Размер описанного массива — 25 элементов:

```
var a: array[1..25] of integer;
```

**Пример 3.6.** Опишем массив, рассмотренный в примере 3.2:

```
var S: array[1..27] of string;
```

**Пример 3.7.** Опишем массив, рассмотренный в примере 3.3. Размер описанного массива — 31 элемент.

```
var T: array[1..31] of real;
```

**Пример 3.8\*.** Описать массив для хранения следующих данных: имеется здание склада, в котором есть два подземных этажа, цокольный и три верхних. Необходимо хранить количество пустых отсеков склада на каждом этаже. Размер описанного массива — 6 элементов:

```
const verh = 3;
      niz = -2;
var Sklad: array[niz..verh] of
                integer;
```

**Пример 3.9.** Команда `b := a;` допустима для массивов `a` и `b`, описанных следующим образом:

```
var a, b: array[1..10] of integer;
```

Но команда `b := a;` выдаст ошибку, если массивы будут описаны так:

```
var a: array[1..10] of integer;
    b: array[1..10] of integer;
```

или так:

```
var a: array[1..10] of integer;
    b: array[1..15] of integer;
```

или так:

```
var a: array[1..10] of integer;
    b: array[1..10] of real;
```

Имя массива является идентификатором и задается по тем же правилам, что и имена любых других переменных.

Тип индекса определяет, как будут нумероваться элементы в массиве. Для задания типа индекса указывают номер первого элемента в массиве, затем ставят две точки, после которых указывают номер последнего элемента. Данные, которые используются для задания индексов, должны быть константами. Диапазон индексов определяет максимально возможное количество элементов в массиве — **размер массива**.

Тип элементов задает значение базового типа для данного массива. Базовый тип может быть любым из известных вам типов (примеры 3.5—3.8).

### 3.3. Операции над массивами

Массивы, описанные одинаково (в одной команде описания), можно использовать в операциях присваивания. В результате выполнения этой команды все элементы одного массива будут переписаны во второй (пример 3.9). Если массивы описаны одинаково, но в разных строках или описаны порозному, то при попытке присваивания возникнет ошибка о невозможности преобразовать типы.

Никакие другие операции для массива как для типа данных не определены.

Операции, выполняемые с элементами массива, соответствуют операциям, выполняемым над базовым типом. Если, например, описан массив из чисел типа `integer`, то с элемента-

ми такого массива можно выполнять такие же операции, как и с целыми числами. Элементы массива называются **индексированными переменными**. Они могут использоваться так же, как и простые переменные (пример 3.10).

### 3.4. Ввод и вывод элементов массива

Чтобы работать с массивом, необходимо задать начальные значения элементов массива. Сделать это можно несколькими способами:

- 1) ввод элементов массива с клавиатуры;
- 2) использование случайных чисел для определения значений;
- 3) использование функций (стандартных или собственных) для определения значений;
- 4) определение элементов массива как констант.

При вводе элементов массива с клавиатуры каждый элемент должен вводиться отдельно. Если количество вводимых элементов определено, то можно воспользоваться циклом **for** (пример 3.11).

При вводе элементов массива следует помнить, что количество вводимых элементов не может быть больше размера массива. В массив, описанный в примере 3.11, можно ввести любое количество чисел от 1 до 10, изменив значение 10 в заголовке цикла.

При описании массива размер определяет максимальное количество возможных элементов. При вводе можно определять количество элементов, которое необходимо для обработки в каждом конкретном случае (пример 3.12).

#### Пример 3.9. Продолжение.

```

var a, b : array [1..10] of integer;
    c: array [1..10] of integer;
begin
  for var i := 1 to 10 do
    a[i] := i;
  b := a;
  for var i := 1 to 10 do
    write(b[i], ' ');
  c := b;

```

**Пример 3.10.** Операции над индексированными переменными:

```

a[3] := 25 mod 7;
s := (t[1] + t[30])/2;
a[k] := b[k]*2;
Sum := Sum + a[i];
if a[i] < 0 then ...

```

**Пример 3.11.** Ввести 10 элементов массива a.

```

var a: array[1..10] of integer;
begin
  writeln('Введите 10 чисел
        через пробел');
  for var i := 1 to 10 do
    read(a[i]);
  ...
end.

```

**Пример 3.12.** Ввести заданное количество элементов массива a.

```

var a: array[1..100] of integer;
    n: integer;
begin
  writeln('Введите количество
        чисел в массиве');
  readln(n);
  writeln('Введите', n,
        'чисел через пробел');
  for var i := 1 to n do
    read(a[i]);
  ...
end.

```

**Пример 3.13.** Ввод массива строк.

```
var a: array[1..100] of string;
    n: integer;
begin
  writeln('Введите количество
    строк в массиве');
  readln(n);
  writeln('Введите', n, 'строк,
    каждую с новой строки!');
  for var i := 1 to n do
    readln(a[i]);
  ...
end.
```

**Пример 3.14.** Случайным образом задать  $n$  элементов массива  $a$ . Каждый элемент — число из отрезка  $[0; 100]$ .

```
var a: array[1..100] of integer;
    n: integer;
begin
  writeln('Введите количество
    чисел в массиве');
  readln(n);
  for var i := 1 to n do
    a[i] := random(101);
  ...
end.
```

**Пример 3.15.** Описание массива, элементы которого являются числовыми константами.

```
const simple _ numb: array[1..5]
  of integer = (2, 3, 5, 7, 11);
```

**Пример 3.16.** Описание массива, элементы которого являются строковыми константами.

```
const c _ rgb: array of string =
  ('красный', 'синий', 'зеленый');
```

**Пример 3.17.** Вывод элементов массива в столбец (по одному в строке).

```
for var i := 1 to n do
  writeln(a[i]);
```

**Пример 3.18.** Вывод элементов массива в строку (через пробел).

```
for var i := 1 to n do
  write(a[i], ' ');
```

Если необходимо вводить массив из строк, нужно помнить, что каждый элемент вводится в отдельной строке с использованием команды `readln` (пример 3.13). Использовать пробел как разделитель не получится, поскольку пробел будет воспринят как очередной символ строки.

Иногда бывает удобно задавать элементы массива случайным образом. Для этого используется функция `random(k)`, которая генерирует случайное целое число из промежутка  $[0; k)$  (пример 3.14).

Если элементы массива должны принадлежать отрезку  $[a; b]$ , то можно использовать функцию `random(a, b)` или определить значение элемента массива так:

$$a[i] := \text{random}(b-a+1) + a;$$

Если элементы массива не будут изменяться при решении задачи, то массив может быть описан как константа (примеры 3.15, 3.16). При таком описании можно не указывать индексы элементов в массиве, тогда нумерация будет осуществляться от нуля до количества элементов в списке минус один.

Выводить элементы массива можно в столбец (пример 3.17) или в строку (пример 3.18). Если элементы массива выводятся в строку, то между ними нужно выводить символ-разделитель (чаще всего используют пробел), иначе все числа будут распечатаны подряд как одно число с большим количеством цифр. Выводить элементы массива можно не только



в прямом, но и в обратном порядке (пример 3.19).

### 3.5. Решение задач с использованием ввода-вывода массивов

**Пример 3.20.** Написать программу, которая введет элементы массива с клавиатуры и выведет сумму третьего и пятого элементов.

Этапы выполнения задания

- I. Исходные данные: массив  $a$  и количество элементов  $n$ .
- II. Результат:  $S$  — сумма третьего и пятого элементов.

III. Алгоритм решения задачи.

1. Ввод исходных данных.
2. Вычисление суммы.
3. Вывод результата.

IV. Описание переменных:  $a$  — `array[1..10] of integer`;  $n$ ,  $S$  — `integer`.

**Пример 3.21.** Написать программу, которая сформирует массив из  $n$  чисел из отрезка  $[0; 100]$  случайным образом. Вывести массив на экран.

Этапы выполнения задания

- I. Исходные данные: массив  $a$  и количество элементов  $n$ .
- II. Результат: полученный массив.
- III. Алгоритм решения задачи.
  1. Ввод исходных данных.
  2. Генерация массива.
  3. Вывод результата.
- IV. Описание переменных:  $a$  — `array[1..100] of integer`;  $n$  — `integer`.

**Пример 3.19.** Вывод элементов массива в строку (в обратном порядке).

```
for var i := n downto 1 do
  write(a[i], ' ');
```

**Пример 3.20.**

V. Программа:

```
var a: array[1..10] of integer;
    n, S: integer;
begin
  writeln('Введите количество
        чисел в массиве >=5');
  readln(n);
  writeln('Введите ', n,
        ' чисел через пробел');
  for var i := 1 to n do
    read(a[i]);
  S := a[3] + a[5];
  write('Сумма чисел = ', S);
end.
```

VI. Тестирование:

Окно вывода

```
Введите количество чисел в массиве
7
Введите 7 чисел через пробел
12 3 4 2 1 19 7
Сумма чисел = 5
```

VII. Анализ результатов. Третий элемент массива равен 4, пятый элемент равен 1, сумма элементов равна 5.

**Пример 3.21.**

V. Программа:

```
var a: array[1..100] of integer;
    n: integer;
begin
  writeln('Введите количество
        чисел в массиве');
  readln(n);
  for var i := 1 to n do
    a[i] := random(101);
  for var i := 1 to n do
    write(a[i], ' ');
  end.
```

VI. Тестирование:

Окно вывода

```
Введите количество чисел в массиве
8
66 44 80 3 100 66 3 78
```

**Пример 3.22.**

V. Программа:

```

var a: array[1..100] of integer;
    n, k: integer;
begin
  writeln('Введите количество
          чисел в массиве');
  readln(n);
  for var i:=1 to n do
  begin
    a[i]:=2*random(10, 35);
    write(a[i], ' ');
  end;
  writeln;
  writeln('Введите k');
  readln(k);
  write(a[k]);
end.

```

VI. Тестирование:

## Окно вывода

```

Введите количество чисел в массиве
7
50 56 66 60 32 24 66
Введите k
5
32

```

**Пример 3.23.**

V. Программа:

```

var s: array [1..20] of string;
    n, k1, k2: integer;
begin
  writeln('Количество учащихся ');
  readln(n);
  writeln('Фамилии');
  for var i := 1 to n do
    readln(s[i]);
    writeln('k1 и k2');
    readln(k1, k2);
  for var i := k1 to k2 do
    writeln(s[i]);
end.

```

**Пример 3.22.** Написать программу, которая сформирует массив из  $n$  четных чисел из отрезка  $[20; 70]$  случайным образом. Вывести на экран  $k$ -й элемент массива.

Этапы выполнения задания

I. Исходные данные: массив  $a$  и количество элементов  $n$ .

II. Результат: искомым элемент.

III. Алгоритм решения задачи.

1. Ввод исходных данных.

2. Генерация массива.

1.1. Чтобы элементы массива были только четными, необходимо каждый полученный элемент умножить на 2.

1.2. Поскольку элементы умножаются на два, границы исходного отрезка нужно уменьшить в два раза.

1.3. Вывод массива по элементам.

3. Ввод значения  $k$  и вывод результата.

IV. Описание переменных:  $a$  — `array[1..100] of integer`;  $n$ ,  $k$  — `integer`.

**Пример 3.23.** Написать программу, которая введет с клавиатуры список фамилий учащихся и выведет из него фамилии с номерами от  $k_1$  до  $k_2$ .

Этапы выполнения задания

I. Исходные данные: массив  $s$  и количество учащихся  $n$ , номера фамилий —  $k_1$  и  $k_2$ .

II. Результат: список заданных фамилий.

III. Алгоритм решения задачи.

1. Ввод исходных данных.

2. Вывод результата.

IV. Описание переменных: `s — array[1..20] of string`; `n, k1, k2 — integer`.

**Пример 3.24.** Задать случайным образом два массива `X` и `Y`, содержащих по `n` чисел из отрезка `[100; 300]`, и массив `R`, содержащий `n` чисел из отрезка `[5; 100]`. Построить на экране окружности, координаты центров которых хранятся в массивах `X` и `Y`, а радиусы в массиве `R`.

Этапы выполнения задания

I. Исходные данные: массивы `X`, `Y`, `R` и количество элементов `n`.

II. Результат: рисунок `n` окружностей.

III. Алгоритм решения задачи.

1. Ввод исходных данных.

2. Генерация массивов.

3. Установка прозрачного стиля заливки для того, чтобы изображались окружности, а не круги.

4. Вывод результата.

IV. Описание переменных: `X, Y, R — array[1..100] of integer`; `n — integer`.

Все рассмотренные выше способы ввода и вывода массивов универсальны и могут использоваться для разных компиляторов языка Pascal. В среде PascalABC.Net дополнительно реализованы команды `print` и `println`, с помощью которых массив можно вывести без использования команды цикла. Команда `print(b)` выведет элементы массива `b` в квадратных скобках через запятую: `[1,2,3,4,5,6]`.

Команда `println` после вывода массива дополнительно переводит курсор на новую строку. Элементы выводятся так же, как и при использовании команды `print`.

**Пример 3.23.** Продолжение.

VI. Тестирование:

```
Окно вывода
Количество учащихся
6
Фамилии
Белов
Иванов
Королев
Петров
Сидоров
Яшкин
k1 и k2
3
5
Королев
Петров
Сидоров
```

**Пример 3.24.**


V. Программа:

```
uses graphABC;
var X, Y, R: array[1..100]
    of integer;
    n: integer;
begin
  SetWindowSize(400,400);
  writeln('Введите количество
    чисел в массиве!');
  readln(n);
  writeln(n);
  for var i := 1 to n do
  begin
    X[i]:= random(100,300);
    Y[i]:= random(100,300);
    R[i]:= random(5,100);
  end;
  SetBrushStyle(bsClear);
  for var i := 1 to n do
    circle(X[i],Y[i],R[i])
  end.
```

VI. Тестирование.





-  1. Что такое массив?  
 2. Как описываются массивы?  
 3. Что такое размер массива?  
 4. Какие операции допустимы для массивов?  
 5. Какие способы задания значений элементам массива вы знаете?  
 6. Как можно вывести массив?

  **Упражнения**

- 1 Используя примеры 3.14—3.18, выполните следующие задания.
- Введите 5 чисел и выведите их в одной строке.
  - Введите 7 чисел и выведите их в одной строке в обратном порядке.
  - Задайте 10 случайных чисел и выведите их по одному в строке.
  - Выведите на экран элементы массива, заданного в примере 3.16.
- 2 Измените программу из примера 3.19 так, чтобы выводилось произведение первых трех элементов.
- 3 Используя программы из примера 3.19 или 3.20, задайте массив из  $n$  случайных чисел из отрезка  $[-10; 10]$ . Выведите: первый элемент; последний элемент; элемент, стоящий на среднем месте.
- 4 Введите массив из  $n$  строк с клавиатуры. Выведите элементы массива в обратном порядке.
- 5 Для массива, описанного в примере 3.2, введите данные с клавиатуры. Задайте номер учащегося. Выведите его фамилию.
- 6\* Введите рост учащихся своего класса, организовав ввод следующим образом:  
 Введите количество учащихся в классе: 15  
 Вводите рост учащихся  
 учащийся номер 1: 165  
 учащийся номер 2: 170  
 учащийся номер 3: 156
- 7\* Для массива, описанного в примере 3.3, задайте значения случайными вещественными числами из интервала  $(-20; 10)$ . Выведите значения температур для указанного диапазона дат. Пример вывода для диапазона дат от 1 декабря до 8 декабря:  
 1 декабря температура была = 9.4  
 2 декабря температура была = -11.8  
 3 декабря температура была = -16.6  
 4 декабря температура была = 8  
 5 декабря температура была = 0.9  
 6 декабря температура была = -9.3  
 7 декабря температура была = -11.5  
 8 декабря температура была = 6.6
- 8 Измените программу из примера 3.24 так, чтобы окружности рисовались разными цветами.