§10. Решение задач с использованием массивов

Рассмотрим алгоритмы решения задач с использованием одномерных числовых массивов на примерах из различных предметных областей.

Решение любой задачи с помощью компьютера содержит несколько этапов:

I. Определение исходных данных (что дано?).

II. Определение результатов (что требуется найти?).

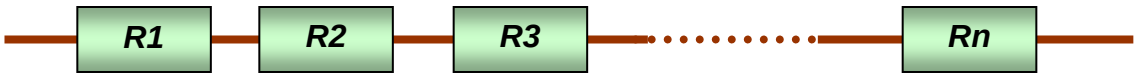
III. Описание переменных (определение типов данных).

IV. Составление алгоритма решения задачи.

V. Написание и отладка программы.

VI. Тестирование программы.

Пример 1. В массив вводятся значения сопротивлений **n** резисторов. Составить программу для расчета общего сопротивления электрической цепи при последовательном соединении этих резисторов (рис. 2.8).



I. Исходными данными являются количество резисторов и сопротивление каждого из них. Будем вводить эти данные с клавиатуры и помещать соответственно в переменную *n* и массив *R*. Индексом массива будет номер резистора от **1** до **n**.

II. Требуется найти общее сопротивление цепи. Будем помещать его в переменную *Rs*.

III. Пусть максимальное количество соединяемых резисторов (размер массива) равно 20, а их сопротивления измеряются в Омах и имеют целочисленные значения. Тогда все переменные имеют тип *integer* и могут быть описаны так: *var R: array[1..20] of integer; var n, i, Rs: integer ;*

IV. Алгоритм решения задачи основан на использовании формулы: общее сопротивление цепи при последовательном соединении равно сумме сопротивлений всех резисторов **Rs = R1 + R2+…+Rn.**

Вычисление суммы производится в цикле *for*. В качестве начального значения имеет смысл взять сопротивление первого резистора *Rs:=R[1]* (подключен один резистор). Тогда значение счетчика цикла *i* изменяется от **2** до *n*. На каждом шаге цикла к цепи подключается еще один резистор (к значению *Rs* прибавляется сопротивление очередного резистора *R[i]*).

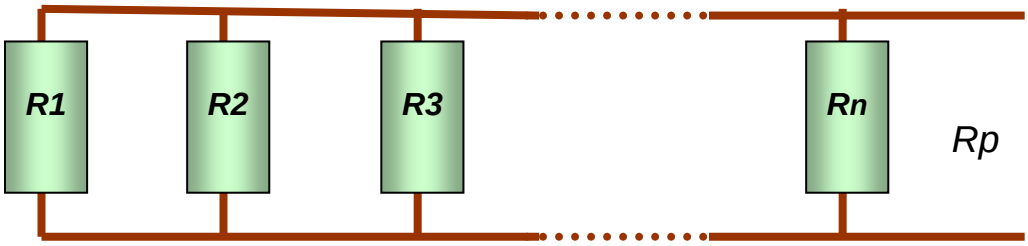
V. Программа имеет вид:

```
program Primer10_1;
var R: array[1..20] of integer ;
    n, i, Rs: integer ;
begin
    write('введите количество резисторов (<=20)  '); readln(n);
    writeln('введите их сопротивления в Омах  ');
       for i:=1 to n do read(R[i]);                    { ввод сопротивлений }
    Rs:= R[1];                                         { задание начального значения }
       for i:=2 to n do
            Rs := Rs + R[i];                           { вычисление сопротивления цепи }
    writeln('общее сопротивление цепи = ', Rs, ' Ом');          { вывод }
end.
```

VI. Протестируем программу. Результат может выглядеть так:

```
введите количество резисторов (<=20)  7
введите их сопротивления в Омах
47 120 51 100 47 75 91
общее сопротивление цепи = 531 Ом
```

Пример 2. В массив вводятся значения сопротивлений **n** резисторов. Составить программу для расчета общего сопротивления электрической цепи при параллельном соединении этих резисторов (рис. 2.9).



Решение этой задачи аналогично предыдущему. Отличие в том, что алгоритм решения основан на использовании формулы расчета общего сопротивления цепи при параллельном соединении резисторов (суммируются обратные величины): **1/Rp = 1/R1 +1/R2+…+1/Rn.**

Переменная *Rp* должна иметь вещественный тип (либо преобразовываться к целому типу одной из функций округления). Описание переменных принимает вид: *var R: array[1..20] of integer; var n, i: integer ; var Rp: real;*

Вычисления производятся в цикле ***for***. В качестве начального значения имеет смысл взять сопротивление первого резистора ***Rp:=R[1]***. Значение счетчика цикла *i* изменяется от **2** до *n*. На каждом шаге цикла к цепи подключается еще один резистор (к значению *1/Rp* прибавляется величина *1/R[i]*, обратная сопротивлению очередного резистора).

Программа имеет вид:

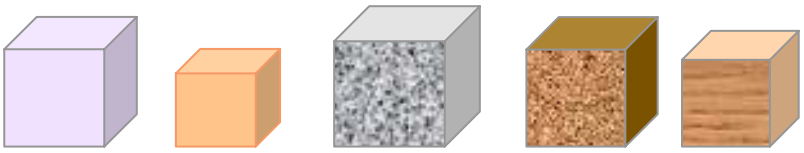
```
program Primer10_2;
var R: array[1..20] of integer ;
    n, i  integer ; var Rp: real ;
begin
    write('введите количество резисторов (<=20)  '); readln(n);
    writeln('введите их сопротивления в Омах');
       for i:=1 to n do read(R[i]);                    { ввод сопротивлений }
    Rp:=R[1];                                          { задание начального значения }
       for i:=2 to n do
            Rp := 1/(1/Rp + 1/R[i]);                   { вычисление сопротивления цепи }
    writeln('общее сопротивление цепи = ', Rp:6:2, ' Ом');        { вывод }
end.
```

Протестируем программу. Результат может выглядеть так:

```
введите количество резисторов (<=20)  7
введите их сопротивления в Омах
47 120 51 100 47 75 91
общее сопротивление цепи =   9.54 Ом
```

Пример 3. В лабораторной работе измеряют массу **m** и ребро **a** пяти образцов кубической формы (рис. 2.10). Составить программу для расчета плотности вещества. Данные измерений вводятся с клавиатуры. Результаты измерений и вычислений выводятся в таблицу

образец	масса, г	ребро, см	плотность, г/см^3.



I. Исходными данными являются масса и размер образцов. Будем вводить эти данные с клавиатуры и помещать соответственно в массивы ***m*** и ***a*** . Индексом массивов будет номер образца от 1 до 5.

II. Требуется вычислить и вывести в таблицу плотность вещества каждого образца. Чтобы не создавать еще один массив, будем производить вычисления непосредственно при выводе в процедуре *writeln()*.

III. Масса измеряется в граммах, а размер в сантиметрах с точностью до одного десятичного знака. (вещественный тип). Таким образом, переменные могут быть описаны так:

```
var m, a : array[1..20] of real;   i: integer;
```

IV. Алгоритм решения задачи основан на использовании формулы плотности вещества $d = m/v = m/a^3$.

Вычисление и вывод результатов в таблицу производятся в цикле ***for***.

V. Программа имеет вид:

```
program Primer10_3;
var m, a : array[1..5] of real;
    i: integer;
begin
    writeln('попарно вводите массу (в г) и ребро (в см) образцов');
    for i:=1 to 5 do readln(m[i], a[i]);                    { ввод m и a }
    writeln('образец : масса : ребро : плотность');      {заголовок таблицы}
    for i:=1 to 5 do
        writeln(i:4, m[i]:10:2, a[i]:9:2, m[i]/(a[i]*a[i]*a[i]):10:2);      { вывод }
end.
```

VI. Протестируем программу. Результат может выглядеть так:

```
попарно вводите массу (в г) и ребро (в см) образцов
67 2.3
36 1.6
85 2.2
45 4.1
52 1.8
образец : масса : ребро : плотность
   1      67.00     2.30      5.51
   2      36.00     1.60      8.79
   3      85.00     2.20      7.98
   4      45.00     4.10      0.65
   5      52.00     1.80      8.92
```

Особый интерес представляет решение задач с использованием числовых массивов и графики. Графические элементы улучшают наглядность и выразительность представляемой информации, а также могут иметь и самостоятельное значение, например, в программах с элементами геометрии, рисования, черчения.

Напомним, что рисование производится в графическом окне и требует подключения модуля GraphAbc.

Пример 4. Задан массив из 7 простых чисел. Составить программу, которая нарисует числовую ось и отметит на ней эти числа кружками.

Исходными данными являются простые числа, заданные в массиве:

```
const  A: array[1..7] of integer = (2, 3, 5, 7, 11, 13, 17);
```

Требуется нарисовать числовую ось и отметить на ней эти числа кружками.

Пусть горизонтальная числовая ось проведена из точки с координатами (20, 200) и имеет 18 делений через 40 пикселей. Нетрудно вычислить координату ее конца 20+18*40 = 740. Деления представляют собой вертикальные отрезки длиной 20 пх (200 пх – 180 пх) с абсциссами *20+40*i* .

Центры кружков лежат на числовой оси (ордината 200) и имеют абсциссы *20 + 40* A[i]*. Пусть их радиус 8 пх, а цвет красный *clRed*.

Программа может выглядеть так:

```
program Primer10_4;
uses GraphAbc;                          { подключение графического модуля }
const  A: array[1..7] of integer = (2, 3, 5, 7, 11, 13, 17);
var i: integer;
begin
    line (20,200, 740,200);                          { рисование оси }
    for i:=0 to 18 do                                { рисование делений }
        line(20+40*i,180, 20+40*i,200);
    setBrushColor(clRed);                            { задание цвета кисти}
    for i:=1 to 7 do circle(20+40*A[i], 200,8);      { рисование кружков }
end.
```

Результат работы будет выглядеть так (рис. 2.11):

Пример 5. В массив вводят результаты соревнований по прыжкам в высоту. Составить программу для отображения их в виде линейной диаграммы

Исходными данными являются количество участников соревнований (не более 20) и результат каждого из них (высота прыжка в см – целое число до 150 см). Будем вводить эти данные с клавиатуры и помещать соответственно в переменную **n** и массив **h**.

Линейная диаграмма представляет собой вертикальные отрезки, высота которых в пикселях равна высоте прыжков *h[i]* в см. Нижние их концы лежат на горизонтальной оси и имеют координаты (*40*i, 200*), а верхние – (*40*i, 200 -- h[i]*). Напомним, что начало отсчета находится в левом верхнем угле экрана, ось абсцисс направлена вправо, а ось ординат – вниз.

При работе с графическим окном системы PascalABC удобно все данные вводить и выводить в этом окне. Совмещать работу с текстом и графикой в одном окне можно, подключив модули CRT и GraphABC одновременно.

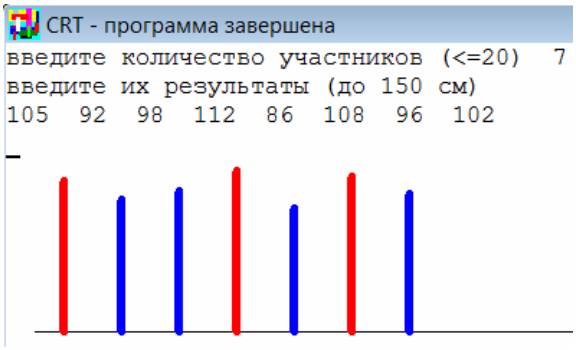
Программа может выглядеть так:

```
program Primer10_5;
uses  CRT, GraphAbc;                          { подключение модулей }
var h : array[1..20] of integer;
      n, i: integer;
begin
    write('введите количество участников (от 5 до 20)  '); readln(n);
    writeln('введите их результаты (до 150 см)   ');
        for i:=1 to n do read(h[i]);           { ввод результатов }
    line (20,200, 740, 200);                   { рисование горизонтальной оси }
    setPenColor(clRed);                        { задание цвета пера }
    setPenWidth(6);                            { задание толщины линии }
          for i:=1 to n do
              line(40*i, 200,  40*i, 200-h[i]);    { рисование вертикальных линий }
end.
```

Результат работы может выглядеть так (рис. 2.12а):

а

б



Несложно изменить эту программу так, чтобы результаты меньше, например, 100 см отображались синим цветом. Изменения заключаются в выборе цвета пера по условию в цикле рисования линий.

```
if h[i]<100 then setPenColor(clBlue) else setPenColor(clRed);
```

Результат работы показан на рисунке 2.12 б.

Упражнения

1. Сосуды цилиндрической формы заполнены до краев жидкостью плотностью d = 1200 кг/м^3 Составьте программу, которая вычисляет одну из величин:

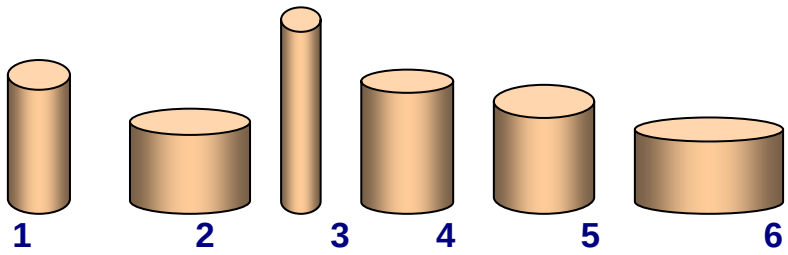
а) давление столба жидкости на дно сосудов p = d g h;

б) массу жидкости в каждом сосуде m = d V = d h S,

с) суммарную массу жидкости,

Требуемые исходные данные (радиус сосуда **R,** высота столба жидкости **h**) вводятся в массив с клавиатуры, результат выводится на экран.

Ускорение свободного падения g = 9.8 м./с^2. Площадь дна вычисляется по формуле $S= 3.14*R^2$



2. В лабораторной работе по определению теплоемкости вещества измеряют приращения температуры **dT** при сообщении одинакового количество теплоты **Q** = 420 Дж шести образцам одинаковой массы **m**=0.12 кг, изготовленным из разных материалов.

Составьте программу, которая вычисляет теплоемкость веществ **c = Q/ m dt.** Данные измерений dT вводятся в массив с клавиатуры. Результаты измерений и вычислений выводятся в таблицу.