**МОУ Селятинская средняя общеобразовательная школа №1**

**Наро - Фоминский район Московской области**

**Массивы в Turbo Pascal. Решение задач на массивы.**

**Проверил: Кащей Владимир Васильевич**

**Выполнил: слушатель курсов «Методика преподавания**

 **алгоритмизации и программирования в**

 **средней школе» учитель информатики**

 **Пархоменко Т.А.**

**2011 г.**

Если работа программы связана с хранением и обработкой большого количества однотипных переменных, для их представления в программе можно использовать массивы.

 Например, пусть программа пользователя выполняет некоторые действия над последовательностью целых чисел, насчитывающей сто элементов i1, i2, …,i100, которые требуется сохранить до конца ее работы. Вместо того чтобы описывать указанные переменные сто раз, можно один раз объявить целочисленную переменную i, состоящую из ста элементов, - массив.

 Массив представляет собой совокупность данных одного типа с общим для всех элементов именем.

 Элементы массива пронумерованы, и обратиться к каждому из них можно по номеру. Номера элементов массива иначе называются индексами, а сами элементы массива – переменными с индексами (индексированными переменными).

Характеристики массива:

* тип – общий тип всех элементов массива;
* размерность – количество индексов массива;
* диапазон изменения индекса (индексов) – определяет количество элементов в массиве.

 Вектор (одномерный массив) – это пример массива, в котором элементы нумеруются одним индексом.

 Если в массиве хранится таблица значений (матрица), то такой массив называется двумерным, его элементы нумеруются двумя индексами – номером строки и столбца соответственно.

 В качестве номера(индекса) элемента массива, в общем случае используется выражение порядкового типа. Наиболее часто индекс – это целая константа или переменная типа integer, реже – типа char или Boolean.

 При обращении к элементу массива индекс указывается в квадратных скобках после имени массива. Например, a[3], b[1,2]. Однако использование элементов массива в качестве обычных переменных не дает существенной выгоды. Массивы ценны тем, что их индексы сами могут быть переменными или выражениями, обеспечивая доступ не к одному, а к последовательности элементов. Обработка массивов производится при изменении индексов элементов.

 Например, в случае использования выражения следующие переменные удобно применять для просмотра в цикле элементов массива:

* a[i] – всех элементов;
* a[2\*i] – элементов, стоящих на четных местах;
* a[2\*i-1] – элементов, стоящих на нечетных местах.

**Описание массива**

 Самый простой способ описания массива – это объявит переменную в разделе описания переменных var с использованием зарезервированного слова array (т.е. массив). В общем виде описание выглядит так:

* для одномерного массива:

var <Имя массива>: array [нижняя граница.. верхняя граница]

 of <Тип элементов>;

Например:

var a: array[1..100] of integer; {100 элементов – целые числа}

var b: array[0..50] of char; {51 элемент – символы }

var c: array[-3..4] of Boolean; {8 элементов – логические значения}

* для двумерного массива:

var <Имя массива>: array [ниж. граница индекс1.. верхняя граница индекс1,

 нижняя граница индекс2.. верхняя граница индекс2]

 of <Тип элементов>;

 При необходимости подсчитать частоту появления в некотором тексте различных букв латинского алфавита, можно воспользоваться вектором счетчиков, индекс которого (латинские буквы) меняется от до , а элементы (целые числа) после подсчета указывают, сколько раз встречается в тексте данная буква.

var count: array [‘a’..’z’] of integer;

 При выполнении программы не обязательно заполнять все ячейки данными, т.е. реальное количество элементов в массиве может быть меньше, чем указано при описании, но ни в коем случае не должно быть больше.

 При объявлении массива нельзя задавать границы индексов при помощи переменных. Память под массив выделяется компилятором до выполнения программы, а переменные получают значения только в ходе ее выполнения.

 Массив также можно описать как типизированную константу в разделе описания констант. Список значений элементов массива при этом заключается в круглые скобки.

const x: array [1..5] of integer=(1,3,5,7,9);

 y: array [1..2,1..3] of integer=(1,3,5),(2,4,6);

 Описание массива как типизированной константы используется на практике:

* для задания массивов с неизменными значениями элементов;
* при отладке программы, чтобы каждый раз не заполнять массив «вручную» при запуске программы.

Следует знать:

* границы индекса должны быть константами;
* нижняя граница индекса чаще всего равна 1;
* при выходе индекса за границы массива возникает программное прерывание, если включена директива компилятора {SR+}. Если эта директива отключена, выход за границы массива может испортить соседние с ним ячейки памяти, что приведет к непредсказуемым ошибкам в программе;
* для ввода, вывода и обработки массивов удобно применять циклы, особенно удобен в этих случаях цикл for, т.к. номера элементов следуют по порядку друг за другом с единичным шагом;
* массивы, идентичные по структуре, т.е. с одинаковыми типами индексов и элементов, могут участвовать в операторе присваивания без указания индекса.

 Массив, описанный как типизированная константа, уже содержит данные. Массивы, объявленные в разделе описания переменных, необходимо заполнить данными, прежде чем выполнять с ними какие-либо действия.

 Значения элементов массива также можно задать следующими способами:

* при вводе данных с клавиатуры;
* с помощью датчика случайных чисел;
* присваиванием заданных значений;
* считывая значения элементов из файла.

*Действия, с одномерными массивами:*

* вычисление суммы элементов;
* вычисление произведения элементов;
* подсчет количества элементов, удовлетворяющих какому-либо условию;
* поиск элемента с заданным значением;
* поиск максимального (минимального) элемента и его номера;
* изменение значений элементов.

*Действия, с двумерными массивами:*

* суммирование элементов каждой строки;
* поиск (максимального) минимального элемента всей матрицы;
* умножение матрицы а на вектор х, в результате получается новый вектор y.

*Сортировка массива.*

 Сортировка – это процесс упорядочивания набора данных одного типа по возрастанию или убыванию значения какого-либо признака.

Следует знать:

* сортировка массивов – одно из наиболее важных действий над массивами в системах сбора и поиска информации, т.к. в отсортированных массивах найти нужную информацию можно гораздо быстрее по сравнению с несортированными;
* существует множество различных алгоритмов сортировки, которые значительно отличаются друг от друга по скорости работы;
* «быстрые» способы сортировки массивов могут дать колоссальный выигрыш на больших массивах, содержащие тысячи элементов, однако для небольших массивов можно использовать самые простые способы сортировки.

Способы сортировки:

1. Линейная сортировка (сортировка отбором)

Идея линейной сортировки по невозрастанию заключается в том, чтобы, последовательно просматривая весь массив, отыскать наибольшее число и поменять его местами с первым элементом. Затем просматриваются элементы массива, начиная со второго, снова находится наибольший, который меняется местами со вторым и т.д.

1. Сортировка методом «пузырька»

Данный метод сортировки основан на том, что в процессе исполнения алгоритма более «легкие» элементы массива постепенно «всплывают». Особенностью данного метода является сравнение, а затем, если нужно, и перестановка соседних элементов.

1. Метод быстрой сортировки с разделением.

В основу сортировки положен метод последовательного дробления массива на части и обмен элементами между частями.

***Примеры различных программ, использующих массивы.***

***Одномерные массивы***

**Задача 1.**

Сформировать одномерный массив из N элементов, где элементы массива- целые случайные числа в пределах от 1 до 45. Напечатать элементы массива в прямом и обратном порядке.

{*Для получения случайных чисел воспользуемся следующей функцией:
Функция Random [(x)]
Формирует случайное число от 0 до X целого или вещественного типа (перед обращением к функции ее целесообразно инициализировать, использовав процедуру Randomize).
X - параметр, указывающий диапазон значений случайного числа. Оно изменяется в пределах 0 до X. Результат в этом случае имеет тип Word (диапазон значений - 0...65535). Если параметр X не задан, результат будет типа Real в пределах 0.0 <Х< 1.0. Для получения массива целых случайных чисел из диапазона от [А,В] random(B-A+1)+A* }
program z1;
  uses crt;
  var i, n:integer;
      a:array[1..10000] of integer;
begin
  clrscr;
  randomize;
  write(' размер =' );readln(n);
  for i:=1 to n do
    begin
{*Получаем случайные числа в пределах от 1 до 45.*}
      a[i]:=random(46))+1;
{*Выводим элементы массива на экран.*}
      write(a[i], ' ' );
    end;
{*Полученный массив печатаем в обратном порядке.*}
  for i:=n downto 1 do
    write(a[i],' ');
  readkey;
end.

**Задача 2.**

Переставить элементы, стоящие на нечетных местах, с соответствующими элементами на четных местах.

program z2;
  uses crt;
  var i, n, r: integer;
  a: array[1.. 10000] of integer;
begin
  clrscr;
  randomize;
  write ('число элементов ');
  readln(n);
  for i:=1 to n do
    begin
{*Получаем случайные числа и выводим их на экран.*}
      a[i]:=random(45)-22;
      write(a[i],' ');
    end;
{*В полученном массиве меняем соседние элементы.*}
  i:=1;
{Пока I<=N-1.}
  while i<=n-1 do
    begin
{*Меняем значения соседних элементов.* }
      r:=a[i];a[i]:=a[i+1];a[i+1]:=r;
{*Увеличиваем индекс на два.*}
      inc(i,2);
    end;
{*Распечатываем измененный массив.*}
  for i:=1 to n do
    write(a[i],' ');
  readkey;
end.

**Задача 3.**

Найти максимальный (минимальный) элемента массива, а также его порядковый номер.

program z3;
  uses crt;
  var i,n,r,max,min,imax,imin:integer;
      a:array[1..10000] of integer;
begin
  clrscr;
  randomize;
  write('число элементов n= ');
  readln(n);
  for i:=1 to n do
    begin
      a[i]:=random(45)-22;
{*Получаем случайные числа и выводим их на экран.*}
      write(a[i],' ');
    end;
{*За начальный максимум (минимум) берем первый элемент массива.*}
  min:=a[1]; max:=a[1];
  for i:=1 to n do
    begin
{*Если найдется элемент, меньший MIN, то MIN будет равен этому элементу. Одновременно запоминаем индекс промежуточного минимума.*}
      if a[i]<=min then
        begin
          min:=a[i]; imin:=i;
        end;
{*Если найдется элемент, больший МАХ, то МАХ будет равен этому элементу. Одновременно запоминаем индекс промежуточного максимума.*}
      if a[i]>= max then
        begin
          max:=a[i]; imax:=i;
        end;
    end;
{*Печатаем минимальный элемент и его индекс.*}
  writeln(min,' номер' , imin);
{*Печатаем максимальный элемент и его индекс.*}
  writeln(max, ' номер' ,imax);
  readkey;
end.

**Задача 4.**

Вычисление суммы положительных элементов массива

program z4;
  const N=10;
  type Mas=fray [1..N] of integer;
  var a: Mas;
      i: integer; { *Счетчик цикла*}
      S:integer; { *Копилка - переменная для суммирования положительных элементов*}
begin
{ *Заполним массив случайными числами в диапазоне -100..+100* }
  randomize;
  for i:=l to N do
    begin
      a[i]:=-100+random(201);
      write(a[i]:5)
    end;
  writeln;
{ *Присвоим переменным начальные значения* }
  S:=0; { *Переменная S - аккумулятор. Она будет накапливать сумму всех положительных элементов. Нужно присвоить ей такое начальное значение, чтобы оно не повлияло на результат суммирования. Таким числом является ноль* }
  for i:=l to N do { *Перебираем все элементы массива* }
    if A[i]>0 then { *Проверяем каждый элемент на положительность* }
      S:=S+A[i]; { *Если элемент положительный, добавляем значение элемента к аккумулятору* }
{ *Выводим результат на экран:* }
  writeln('Сумма положительных элементов =',S);
  readln
end.

**Задача 5.**

Подсчитать количество четных элементов массива, заданного датчиком случайных чисел.

program z5;
  uses crt;
  var i,k,,r:integer;
      a:array[1..10000] of integer;
begin
  clrscr;
  randomize;
  write(' число элемен. п=');readln(n);
  for i:=1 to n do
    begin
      a[i]:=random(45)-22;
      write(a[i], ' ');
{*Проверяем на четность и считаем количество четных элементов.*}
      if a[i] mod 2=0 then inc(k, 1);
    end;
  write('k= ',k);
  readkey
end.

***Двумерные массивы***

**Задача 1.**

Найти сумму всех элементов двумерного массива и сумму элементов каждой строки.

program z1;
  uses crt;
  type mas=array[1..100, 1..100] of integer;
  var a:mas;
      i,j,n,s,sl :integer;
begin
  clrscr;
  randomize;
  write('n=');readln(n);
  for i:=1 to n do
    begin
      for j:=1 to n do
        begin
{*Получаем случайные значения элементов матрицы.*}
          a[i,j]=random(45)-22;
          write(a[i,j]:4);
{*Находим сумму элементов.*}
          s:=s+a[i,j]; sl:=sl+a[i,j];
        end;
{*Печатаем сумму всех элементов каждой строки и обнуляем значение суммы.*}
      writeln('сумма строки =',s);s:=0;
      writeln;
    end;
  writeln('сумма всех элем. sl=',sl);
  readln;
end.

**Задача 2.**

Найти максимальный элемент каждой строки массива и его индексы (всего массива и его индексы).

{Так как элементы могут повторяться, то договоримся, что будем запоминать только индексы первого максимального элемента.}
program z2;
  uses crt;
  type mas=array[1..100,1..100] of integer;
  var a:mas;
      i,j,n: integer;
      max,min,i1,j1,i2,j2:integer;
begin
  clrscr;
  randomize;
  write('n=');readln(n);
  for i:=1 to n do
    begin
{*Так как тип массива integer, то за начальные значения возьмем.*}
      max: =-32 768;
      for j:=1 to n do
        begin
{*Получаем случайные значения элементов матрицы.*}
          a[i,j]:=random(45)-22;
{*Выводим элементы матрицы на экран.*}
          write(a[i,j]:4);
{*Находим максимальный элемент в каждой строке и его индексы.*}
          if a[i,j]>max then
            begin
              max:=a[i,j];i1 :=i; j1 :=j;
            end;
        end;
{*Печатаем максимальный элемент в каждой строке и его индекс.*}
      write (' тах=',тах, ' строка=',i1,' cmon6eц =',j1);
      writeln;
    end;
  readln;
end.
{*Для нахождения максимального элемента всего массива необходимо:
- перенести начальный максимум на одну строку выше;
- перенести печать максимального элемента на две строки вниз.*}

**Задача 3.**

Найти количество элементов, больших некоторого заданного числа X в каждой строке массива (во всем массиве).

program z3;
  uses crt;
  var a: =array[1.. 100,1..100] of integer;
      i,j,n:integer;
      k,l,x:integer;
begin
  clrscr;
  randomize;
  write('n=');readln(n);
{*Задаем значение Х.*}
  x:=0;
  for i:=1 to n do
    begin
      k:=0;l:=0;
      for j:=1 to n do
        begin
          a[i,j]:=random(45)-22;
          write(a[i,j]:4);
{*Считаем число элементов, удовлетворяющих условию задачи.*}
          if a[i,j]>x then k:=k+1 else l:=l+1
        end;
{*Если находим для всего массива, то следующую строку надо убрать, а начальные значения K=0:L=0 перенести выше на одну строку.*}
      write('k=',k, 'l=',l);k:=0;l:=0;;
    end;
{*Печатаем число элементов, удовлетворяющих условию задачи во всем массиве.*}
  write('k=',k, 'l=',l);
  readln;
end.

**Задача 4.**

Определить, является ли данный квадратный массив симметричным относительно своей главной диагонали.

{*Если массив является симметричным, то для него выполняется равенство a[i,j]=a[j,i] для всех i=l,..., n и j=l,..., n при условии, что i>j. Но если встретится хотя бы одна такая пара, что соответствующие элементы не будут равны, то массив будет несимметричным* }
program z4;
  uses crt;
  var a: array[1..100,1..100] of integer;
      i,j,n:integer;
begin
  clrscr;
  randomize;
  write('n='); readln(n);
  for i:=1 to n do
    begin
      for j:=1 to n do
        begin
          a[i,j]:=random(45)-22;
          write(a[i,j]:4);
        end;
      writeln;
    end;
  readln;
  for i:=1 to n do
  for j:=1 to n do
    if (i>j) and (a[i,j]<>a[j,i] then
    begin
      writeln ('no');
      exit;
    end;
  writeln('yes');
  readln;
end.

**Задача 5.**

Удалить строку с номером k.

{*Для того, чтобы удалить строку с номером к, необходимо: - Сдвинуть все строки, начиная с данной, на одну вверх. - Последнюю строку "обнулить", то есть всем элементам последней строки присвоить нулевое значение. Будем выводить на экран сначала все строки, а второй раз, после удаления, на одну меньше.*}
program z5;
  uses crt;
  type mas=array[1..100,1.. 100] of integer;
  var a:mas;
      i,j,k,n:integer;
begin
  clrscr;
  randomize;
  write('n=');readln(n);
{*Создаем и распечатываем двумерный массив.*}
  for i:=1 to n do
    begin
      for j:=1 to n do
        begin
          a[i,j]:=random(45)-22;
          write(a[i,j]:4);
        end;
      writeln;
    end;
{*Вводим номер удаляемой строки.*}
  write('k=');readln(k);
{*Сдвигаем строки на одну вверх, начиная с данной.*}
  for i: =K to n-1 do
  for j:=1 to n do a[i,j]:=a[i+1,j];
{*Обнуляем последнюю строку.*}
  for j:=1 to n do a[n,j]:=0;
{*Печатаем новый массив, в котором на одну строку меньше.*}
  for i:=1 to n-1 do
    begin
      for j:=1 to n do
        begin
          write(a[i,j]:4);
        end;
      writeln;
    end;
  readln;
end.

**Задача 6.**

Поменять местами строки с номерами К1 и К2.

program z6;
  uses art;
  type mas=array[1..100,1..100] of integer;
  var a:mas;
      i,j,n,k1,k2,r: integer;
begin
  clrscr;
  randomize;
  write('n=');readln(n);
{*Создаем и распечатываем двумерный массив.*}
  for i:=1 to n do
    begin
      for j:=1 to n do
        begin
          a[i,j]:=random(45)-22;
          write(a[i,j]:4);
        end;
      writeln;
    end;
{*Вводим номера строк, которые будем менять местами.*}
  write('stroki k1=k2=');readln(k1,k2);
{*Меняем значения К1 и К2 строк между собой.*}
  for j:=1 to n do
    begin
      r:= a[k1,j];a[k1,j]:=:=a[k2,j];
      a[k2,j]:=r;
    end;
{*Распечатаем измененный массив.*}
  for i:=1 to n do
    begin
      for j:=1 to n do
        begin
          write(a[i,j]:4);
        end;
      writeln;
    end;
  readln;
end.

**Задача 7.**

Заполнить массив А размером п\*m следующим образом, например, n=5 и m=5:
 1       2       3       4       5
10      9       8       7       6
11    12     13     14     15
20     19     18     17     16
21     22     23     24     25
То есть заполняется в виде "змейки".

{*Для того, чтобы заполнить, надо вывести правило заполнения, а оно в данном случае будет таким: если ряд нечетный (то есть номер строки - нечетное число), то A [i,j]=(i-1)\*m+j, иначе (то есть когда строка четная) A[i,j]==i\*m-j+l.
По этому правилу и составляем процедуру заполнения.* }
program z7;
  uses crt;
  type mas=array[1..100,1.. 100] of integer;
  var a:mas;
      i,j,n,m:integer;
begin
  clrscr;
  write('n=m=');readln(n,m);
  for i:=1 to n do
    begin
      for j:=1 to m do
        begin
{*Заполняем массив по выведенному правилу заполнения и выводим его на экран.*}
          if i mod 2 =1 then
            a[i,j]:=(i-1)\*m+j
            else a[i,j]:=i\*m-j+1;
          write(a[i,j]:4);
        end;
      writeln;
    end;
  readln;
end.

**Список используемой литературы**

1. Г.Г. Рапаков, С.Ю. Ржеуцкая “Turbo Pascal для студентов и школьников“, Санкт-Петербург, «БХВ-Петербург», 2011г.
2. А.И. Гусева “ Учимся программировать: Pascal 7.0”, Москва, «Диалог-МИФИ», 2011г.
3. С.В. Вольский, П.А. Дмитриев “Turbo Pascal 7.0 для студентов и школьников“, Санкт-Петербург, «Наука и Техника», 2007г.
4. Д.М. Ушаков, Т.А. Юркова “Паскаль для школьников”, Москва-Санкт-Петербург, «ПИТЕР», 2008г.
5. Е.Р. Алексеев “Турбо Паскаль 7.0”, Москва, NT Press, 2006г.