Лабораторная работа №6

Массив. Обработка массива. Поиск в массиве

**Массив -** это пронумерованная последовательность **однотипных** данных, объединенных одним именем. Элементы массива располагаются в последовательных ячейках памяти и определяются индексом (порядковый номер элемента в массиве).

Так как массивы по размеру (количеству элементов) и размерности (одномерные, двумерные, трехмерные и т.д.) могут сильно отличаться друг от друга, то стандартного типа массив не предусмотрено. Он конструируется программистом в разделе описания типов Type (см.*Таблица 1*).

*Таблица 1*

|  |  |
| --- | --- |
| *ОДНОМЕРНЫЙ МАССИВ* | *ДВУМЕРНЫЙ МАССИВ (МАТРИЦА)*  Индекс элемента |
| *Абстрактное представление*  Индекс элемента | |
| Элемент массива   |  |  |  |  |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | |  | *1* |  | *3* | *4* | *5* | *6* | *7* | *8* | *9* | *10* | | ***А*** | 2.7 |  |  |  |  |  |  |  |  |  |   Индекс элемента  Элемент массива | |  |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | --- | |  |  | *1* | *2* | *3* | *4* | *5* | |  | *1* |  | ‘F’ |  |  |  | | ***В*** | *2* |  |  |  |  |  | |  |  |  |  |  |  |  | |
| **Описание массива**   * *С предварительным описанием типа:*   **TYPE**  <*имя типа*>= **ARRAY**[<*диапазон индексов*>] **OF** <*тип элементов*>;  только порядкового типа!  Любой тип, кроме файлового! | |
| CONST N=10;  **TYPE**  **Arr = array [1..N] of real**;  VAR **A,D**: **Arr**; | CONST N=3,m=5;  **Type**  **Matr= array[1..n,1..m] of char;**  Var **B: Matr**; |
| * *Без предварительного описания типа:* | |
| **Var** A:array [1..10]of real; | **Var** B:array[1..3,1..5]of char |
| **Доступ к элементу массива** осуществляется через указание имени массива и индекса элемента: | |
| **A[1]:=2.7;**  **Readln(A[i]);**  **Write(A[i+1]);** | **B[1,2]:=’F’;**  **B[i,j]:=sin(i+j);**  **Write(B[2,j]);** |
| **Ввод/вывод массива производится поэлементно,**  т.е. каждый элемент массива отдельно**:** | |
| Заполнение массива  program Z1;  Const n=10;  type  Mass=array[1..n] of real;  Var A:Mass; i:byte;  Begin  for i:=1 to n do  begin  Write(‘Введите число’);  Readln(A[i]);  end;  readln;  end. | Для заполнения или вывода двумерного массива необходимо два цикла:  program Z2;  type  Matr=array[1..3,1..5]of Byte;  Var B:Matr; i,j:byte;  Begin  for i:=1 to 3 do  for j:=1 to 5 do  begin  B[i,j]:=i+j;  Write(B[i,j]);  End;  end. |
| **Для заполнения массива случайными числами** необходимо сначала подключить генератор случайный чисел**:**  **Randomize;** *{подключение генератора случайных чисел}*  for i:= 1 to 20 do  А[i]:= **random()**; | |

**Последовательная обработка элементов массива**

В задачах с использованием массивов чаще всего требуется последовательная обработка всех элементов массива. Например, чтобы заполнить массив данными или вывести элементы на экран, в файл. Задачи поиска элемента в массиве и задачи упорядочивания элементов массива по возрастанию или убыванию также требуют последовательного перебора всех элементов, и даже не одного. Поэтому обработка массивов чаще всего выполняется в цикле.

Пример 1

Разработать программу определения суммы положительных элементов целочисленного массива А (n), (N<=40).

Const n=40;

**Type TMas=array[1..n]of integer;**

Var **a:TMas**;

m,i:byte; s:integer;

Begin

writeln('введите кол-во элементов массива:');

readln(m);

*{заполнение массива случайными числами и их вывод}*

writeln('исходный массив:');

**for i:=1 to m do**

begin

a[i]:=random(10)-5;

write(a[i]:4);

end;

writeln;

*{подсчет суммы*}

S:=0;

**for i:=1 to m do**

if a[i]>0 then s:=s+a[i];

writeln('сумма=',s:8:2);

End.

Обработка двумерных массивов требует вложенных циклов – один цикл (чаще всего, внешний) перебирает номера строк, другой – номера столбцов.

Пример 2

Дан двумерный массив В(n×m). Посчитать количество отрицательных элементов в каждой строке массива.

Program Otr\_elem\_diag;

Uses crt;

Const n=5; m=4;

Type Matr=array[1..n,1..m]of integer;

Var a:Matr;

i,j,kol:byte;

Begin

*{заполнение массива случайными числами и их вывод}*

**For i:=1 to n do**

begin

**For j:=1 to m do**

begin

A[i,j]:=random(100);

Write(A[i,j];4);

End;

Writeln;

End;

*{подсчет отрицательных элементов в каждой строке}*

For i:=1 to n do

begin

Kol:=0;

**For j:=1 to m do**

If a[i,j]<0 then kol:=kol+1;

Writeln(‘otr. Elem=’,kol);

End;

End.

**Поиск элемента массива по заданному признаку**

Алгоритмы поиска, к которым также можно отнести и алгоритм поиска максимума, требуют последовательного просмотра элементов массива. Но, если в алгоритме поиска максимума необходимо использовать цикл с параметром For, так как просматриваются все элементы, то в алгоритме простого поиска элемента по какому-либо признаку эффективнее использовать итерационные циклы.

***Пример 3***

Написать программу, которая определяет первый отрицательный элемент в массиве целых чисел.

Для поиска первого отрицательного элемента в массиве необязательно просматривать весь массив ведь искомый элемент может оказаться на самой первой позиции. Поэтому воспользуемся условным циклом While. Будем проверять элементы до тех пор, пока не встретим искомый элемент, в нашем случае отрицательный.

**i:=1;**

While **(A[i]>=0) and (i<=n)** do

i:=i+1;

Второе условие цикла необходимо, чтоб исключить зацикливание, если отрицательного элемента в массиве нет. Если отрицательный элемент в массиве все же присутствует, то переменная i будет хранить номер позиции этого элемента.

**Перестановка элементов массивов**

Для перемещения (перестановки) двух элементов необходима дополнительная переменная (буферная), в которую временно копируется первый элемент перестановки, чтобы на его место записать второй элемент перестановки. А из буфера первый элемент копируется на место второго элемента (см. *Рисунок 1*).

Пример 4

Дан линейный массив из N целых чисел. Поменять местами первый и последний элементы массива.

*Исходный массив:*

**-5**

**Buf:=A[1];**

**A[1]:=A[n];**

**A[n]:=buf;**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | *1* | *2* | *3* | *4* | *5* | *6* | *7* | *8* | *9* | *10* |
| А: | **6** | -9 | 0 | 13 | 4 | 5 | 6 | 12 | 3 | **-5** |

**6**

6

**6**

Buf:

*Результат:*

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | *1* | *2* | *3* | *4* | *5* | *6* | *7* | *8* | *9* | *10* |
| А: | **-5** | -9 | 0 | 13 | 4 | 5 | 6 | 12 | 3 | **6** |

*Рисунок 1*

**Вставка/удаление элемента массива**

**Удаление** элемента массива осуществляется за счет сдвига элементов, расположенных после удаляемого элемента, влево на одну позицию (см. *Рисунок 2*).

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | *1* | *2* | *…* | *…* | ***k*** | ***k+1*** | *…* |  |  | *n*  Для сдвига элементов понадобится цикл:  For **i:=k** to **n-1** do  **A[i]:=A[i+1]**; |
| А: | -5 | -9 | 0 | 13 | 4 | 5 | 6 | 12 | 3 | 6 |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | *1* | *2* | *…* | *…* | ***k*** | ***k+1*** | *…* |  | *n-1* | *n* |
| А: | -5 | -9 | 0 | 13 | **5** | **6** | **12** | **3** | **6** | 6 |

*Рисунок 2*

При этом длина массива как бы становится короче, т.е. n-1.

**Вставка** нового элемента в массив выполняется аналогично за счет сдвига элементов массива вправо на одну позицию:

For i:=**n** downto **k+1** do

**A[i]:=A[i-1]**;

**A[k]:=elem;**

При этом в массиве должна быть свободная ячейка.

**Задания для самостоятельной работы**

Задание 6.1

Заполнить массив числами по заданному правилу:

1. Дано целое число *N* (> 0). Сформировать и вывести целочисленный массив размера *N*, содержащий *N* первых положительных нечетных чисел: 1, 3, 5, … .
2. Дано целое число *N* (> 0). Сформировать и вывести целочисленный массив размера *N*, содержащий степени двойки от первой до *N*-й: 2, 4, 8, 16, … .
3. Заполнить одномерный массив А(n) элементами по следующему правилу: a1=1, a2=2, ai=ai-1+ai-2.
4. Построить прямоугольную таблицу А целых чисел, состоящую из 5 строк и 6 столбцов по правилу: aij=i+2j. Вывести полученную таблицу на печать.
5. Даны целые числа a1, a2, а3. Получить прямоугольную таблицу B из 3 строк и 3 столбцов по правилу: bij=ai-3aj. Вывести полученную таблицу на печать.
6. Дано натуральное число n. Получить прямоугольную таблицу А, состоящую из n строк и n столбцов по правилу:



Задание 6.2

Дан линейный массив целых чисел M(N). Вычислить:

1. Сумму элементов массива, больших числа b (число b вводится с клавиатуры).
2. Количество отрицательных элементов массива.
3. Среднее арифметическое значение элементов массива.
4. Количество элементов, принадлежащих интервалу [a, b] (a, b вводятся с клавиатуры).
5. Количество четных элементов массива.
6. Среднее арифметическое положительных элементов массива.
7. Сумму элементов массива, стоящих на нечетных позициях.
8. Сумму первых пяти отрицательных элементов массива.

Задание 6.3

Дан линейный массив целых чисел M(N). Определить:

1. Позицию (индекс) элемента, равного значению b. Если таких элементов несколько, вывести позицию первого.
2. Максимальный элемент массива и его позицию, при условии, что все элементы различны.
3. Номера всех элементов массива с максимальным значением.
4. Минимальный элемент массива.
5. Минимальный элемент из его элементов с четными номерами: A2, A4, A6, … .
6. Есть ли в данном массиве два соседних четных элемента? Найти номера первой пары.
7. Пару соседних элементов с суммой, равной заданному числу *b*, вводимому в клавиатуры.
8. Есть ли в массиве элементы, являющиеся степенью двойки? Если есть, то вывести их номера. Воспользоваться функцией.
9. Есть ли в массиве простые элементы. Вывести их позиции. Воспользоваться функцией.

Задание 6.4

Дан массив размера *N*. Найти номер его первого локального минимума (*локальный минимум* — это элемент, который меньше любого из своих соседей).

Задание 6.5

Дана прямоугольная таблица n×n. Найти произведение элементов таблицы, расположенных на главной диагонали[[1]](#footnote-1).

Задание 6.6

Дана прямоугольная таблица m×m. Найти сумму элементов таблицы, расположенных выше главной диагонали.

Задание 6.7

Дан массив А(15). Максимальный и минимальный элементы массива поменять местами. Вывести исходный и результирующий массивы.

Задание 6.8

Дана прямоугольная таблица из n строк и m столбцов целых чисел. Найти количество положительных элементов в каждом столбце этой таблицы.

Задание 6.9

Дан массив А(15) целых чисел. Удалить из массива все отрицательные элементы.

Задание 6.10

Дан целочисленный массив размера *N*, не содержащий одинаковых чисел. Проверить, образуют ли его элементы *арифметическую прогрессию* (см. задание Array3). Если образуют, то вывести разность прогрессии, если нет — вывести 0.

Задание 6.11

Дана квадратная матрица A(n,n), n≤10. Разработать программу, которая меняет местами элементы главной и побочной диагоналей. Вывести на печать исходную матрицу и матрицу-результат.

Задание 6.12

Разработать программу, которая определяет строку, содержащую минимальный элемент матрицы n×m, и выполняет перестановку данной строки и первой строки матрицы.

Задание 6.13

Найти суммы элементов строк матрицы В (8,8). Выбрать из них наибольшую. Строку с наибольшей суммой элементов занулить. Напечатать исходную и полученную матрицы.

Задание 6.14

Переписать первые элементы каждой строки матрицы А (10,10) в массив В, если в строке есть элемент больше 13. Если в строке нет таких элементов, то записать 0.

Задание 6.15

Сожмите одномерный массив, удалив элементы, предшествующие минимальному элементу.

Задание 6.16

Дан линейный массив целых чисел. Удалить из массива элементы, меньшие среднего арифметического элементов массива.

Задание 6.17

Дан целочисленный массив размера N. Удалить из массива все элементы, встречающиеся ровно два раза, и вывести размер полученного массива и его содержимое.

Задание 6.18

Дан массив размера *N*. Вставить элемент с нулевым значением перед минимальным и после максимального элемента массива.

Задание 6.19

Дан массив размера N. После каждого отрицательного элемента массива вставить элемент с нулевым значением.

Задание 6.20

Разработать программу, которая удаляет из массива элементы, начиная с n-го индекса по k-ый индекс. Номера индексов вводятся с клавиатуры.

Задание 6.21

Дана действительная матрица А размером N×M. Сформировать одномерный массив В, в котором элемент В(i)=1, если элементы i-ой строки матрицы составляют строго возрастающую последовательность и 0 – в противном случае.

Задание 6.22

Дан массив размера N. Найти количество участков, на которых его элементы убывают.

Задание 6.23

Дан массив размера N. Найти количество его промежутков монотонности (т. е. участков, на которых его элементы возрастают или убывают).

Задание 6.24

Дано число R и массив A размера N. Найти элемент массива, который наиболее близок к числу R (т. е. такой элемент AK, для которого величина |AK − R| является минимальной).

Задание 6.25

Дан целочисленный массив размера N, все элементы которого упорядочены (по возрастанию или по убыванию). Найти количество различных элементов в данном массиве.

Задание 6.26

Дан целочисленный массив размера N. Если он является перестановкой, т. е. содержит все числа от 1 до N, то вывести 0; в противном случае вывести номер первого недопустимого элемента.

Задание 6.27

Дан целочисленный массив *A* размера *N*. Назовем *серией* группу подряд идущих одинаковых элементов, а *длиной серии* — количество этих элементов (длина серии может быть равна 1). Сформировать два новых целочисленных массива *B* и *C* одинакового размера, записав в массив *B* длины всех серий исходного массива, а в массив *C* — значения элементов, образующих эти серии.

Задание 6.28

Дан целочисленный массив размера N. Вставить перед каждой его серией элемент с нулевым значением.

Задание 6.29

Дано множество *A* из *N* точек (точки заданы своими координатами *x*, *y*). Среди всех точек этого множества, лежащих в первой или третьей четверти, найти точку, наиболее близкую к началу координат. Если таких точек нет, то вывести точку с нулевыми координатами.

Задание 6.30

Дано множество *A* из *N* точек (*N* > 2, точки заданы своими координатами *x*, *y*). Найти наибольший периметр треугольника, вершины которого принадлежат различным точкам множества *A*, и сами эти точки (точки выводятся в том же порядке, в котором они перечислены при задании множества *A*).

Задание 6.31

Дан массив размера *N*. Переставить в обратном порядке элементы массива, расположенные между его минимальным и максимальным элементами, включая минимальный и максимальный элементы.

***ИСТ: №6.1(b), 6.2(f), 6.7, 6.3(h), 6.10***

***Допол: 6.16, 6.17, 6.23, 6.27***

1. Главная диагональ квадратной матрицы n×n соединяет верхний левый угол и правый нижний. [↑](#footnote-ref-1)