

ДВУМЕРНЫЕ МАССИВЫ

- Массивы, положение элементов в которых описывается двумя индексами, называются **двумерными.**
- Их можно представить в виде прямоугольной таблицы или матрицы.

**Рассмотрим матрицу A
размером 2×3 ,
то есть в ней будет две строки,
а в каждой строке по три элемента:**

$$A = \begin{Bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{Bmatrix}$$

- Каждый элемент имеет свой номер, как у одномерных массивов, но сейчас номер уже состоит из двух чисел – номера строки, в которой находится элемент, и номера столбца.
- Таким образом, номер элемента определяется пересечением строки и столбца.
- Например, a_{12} – это элемент, стоящий в первой строке и во втором столбце.

Существует несколько способов объявления двумерного массива.

■ **Способ 1.** Двумерный массив можно описать как одномерный, элементами которого являются одномерные массивы. Например, для матрицы A, приведенной выше:

```
Const n=2; m=3;
```

```
type mas1=array[1..m] of <тип эл-тов, напр-р,  
real>;
```

```
    mas2=array[1..n] of mas1;
```

```
var v: mas1; a: mas2;
```

В данном случае переменная v объявлена как одномерный массив из 3-х элементов вещественного типа. Переменная a описана как двумерный массив из 2-х строк, в каждую из которых включено по три элемента

Способ 2.

- Описание массива A можно сократить, исключив определение типа mas1 в определении типа mas2:

```
const n=2; m=3;
```

```
type massiv=array[1..n] of [1..m] of <тип элементов>;
```

```
var a: massiv;
```

Способ 3.

- Еще более краткое описание массива A можно получить, указывая имя массива и диапазоны изменения индексов для каждой размерности массива:

```
const n=2; m=3;
```

```
type massiv=array[1..n,1..m] of <тип элементов>;
```

```
var a: massiv;
```

- Если указанный тип используется для определения одного массива в программе, то удобно объявление массива в разделе описания переменных:

```
var a: array[1..n,1..m] of <тип эл-тов>
```

- Поскольку положение элемента в двумерном массиве описывается двумя индексами (первый – номер строки, второй – номер столбца), программы большинства матричных задач строятся на основе вложенных циклов.
- Обычно внешний цикл работает по строкам матрицы, то есть с его помощью выбирается требуемая строка матрицы, а внутренний цикл – по столбцам матрицы, то есть здесь выбирается нужный элемент из выбранной уже строки.

- **Формирование двумерного массива можно осуществлять всеми тремя способами, описанными для одномерных массивов, то есть**
 - ✓ ввод с клавиатуры,
 - ✓ через генератор случайных чисел или
 - ✓ с помощью формулы.

**Предположим, нам нужно пробежать
всю матрицу.**

- Вот фрагмент программы, выводящий матрицу на экран:

```
for i:=1 to n do  
begin  
    for j:=1 to m do  
    write (a[i,j]:4);  
    writeln  
end;
```

Задача 1.

**Найти сумму всех элементов
двумерного массива.**

```
Program prim1;
type mas=array[1..100,1..100] of integer;
var a:mas; i,j,n,m,s:integer;
begin
  randomize;
  write (`n=`); readln (n); write (`m=`); readln (m);
  for i:=1 to n do
    begin
      for j:=1 to m do
        begin
          a[i,j]:=random(45)-22; write (a[i,j]:4);
          s:=s+a[i,j];
        end;
      writeln;
    end;
  writeln (`summa vsech elem. =`, s);
  readln;
end.
```

Задача 2.

В предыдущей задаче найти не только сумму всех элементов, но и сумму элементов каждой строки.

```
Program prim1;
type mas=array[1..100,1..100] of integer;
var a:mas;
    i,j,n,m,s,s1:integer;
begin
    randomize;
    write (`n=`); readln (n); write (`m=`); readln (m);
    for i:=1 to n do begin
        for j:=1 to m 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]; s1:=s1+a[i,j];
        end;
        {печатаем сумму всех эл-тов каждой строки и обнуляем значение суммы}
        writeln(`summa stroki =`,s); s:=0;
        writeln;
    end;
    writeln (`summa vsech elem. s1=`, s1);
    readln;
end.
```

Задача 3:

- **Найти максимальный элемент каждой строки массива и его индексы.**
- **Найти максимальный элемент всего массива и его индекс (самостоятельно)**

```
Program prim4;
type mas=array[1..100,1..100] of integer;
var a: mas;
    i,j,n,max,i1,j1: integer;
begin
  randomize;
  write ('n='); readln (n);
  for i:=1 to n do begin
    {Так как тип массива integer, то за начальное значение max возьмем}
    max:=-32768;
    for j:=1 to n do begin
      a[i,j]:=random(45)-22;
      write (a[i,j]:4);
      if a[i,j]>max then
        begin
          max:=a[i,j]; i1:=i; j1:=j;
        end;
    end;
    write ('max=', max, 'N stroki=', i1, 'N stolbza=', j1);
    writeln;
  end;
  readln;
End.
```

Задача 4:

- Найти минимальный элемент матрицы и его порядковый номер.

Задача 5:

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

```
Program prim5;  
var a:=array[1..100,1..100] of integer;  
    i,j,n,k,x: integer;  
Begin  
    randomize;  
    write (`n=`); readln (n); write (`x=`); readln (x);  
    for i:=1 to n do begin  
        k:=0;  
        for j:=1 to n do begin  
            a[i,j]:=random(45)-22;  
            write (a[i,j]:4);  
            if a[i,j]>x then k:=k+1;  
        end;  
    end;
```

{Если находим для всего массива, то следующую строку надо убрать, а начальное значение k:=0 перенести выше на одну строку}

```
    writeln(`k=`,k);  
end;  
{Печатаем число элементов, удовлетворяющих условию задачи во всем массиве}  
    write(`k=`,k);  
    readln;  
End.
```

Задача 6:

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

Решение:

- Если массив является симметричным, то для него выполняется равенство $a[i,j]=a[j,i]$ для всех $i=1,\dots,n, j=1,\dots,n$ при условии, что $i>j$. Но если встретится хотя бы одна такая пара, что соответствующие элементы не будут равны, то массив будет несимметричным.

```
Program prim6;
var a: array[1..100,1..100] of integer;
    i,j,n: integer;
begin
    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 (^jes');
    readln;
End.
```

Задача 7:

- Вставить строку из нулей на место строки с номером k .

Решение:

Для решения этой задачи необходимо:

- Первые $k-1$ строк оставить без изменения.
- Все строки после k -й сдвинуть на одну назад, это лучше начать с последней строки и идти до k -й.
- Элементам строки k присвоить заданное значение.
- Кроме того, необходимо изменить размер массива. Так как мы вставляем строку, то число строк будет на одну больше.

```

Program prim7;
type mas=array [1..100,1..100] of integer;
var a:mas; i,j,k,n: integer;
begin
  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 вставляемой строки}
  write (^k='); readln (k);
    {Сдвигаем строки назад до k-й строки, начиная с последней}
  for i:=n downto k do
    for j:=1 to n do a[i+1,j]:=a[i,j];
    {Обнуляем элементы k-й строки}
    for j:=1 to n do a[k,j]:=0;
    {Распечатываем новый массив, который на одну строку больше исходного}
  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 8:

- Удалить строку с номером k .

Решение:

Для того, чтобы удалить строку с номером k , необходимо:

- Сдвинуть все строки, начиная с данной, на одну вверх.
- Последнюю строку «обнулить», то есть всем элементам последней строки присвоить нулевое значение.
- Необходимо выводить на экран сначала все строки, а второй раз, после удаления, на одну меньше.

```
Program prim8;
type mas=array[1..100,1..100] of integer;
var a:mas; i,j,k,n: integer;
begin
  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 9:

- Поменять местами строки с номерами k_1 и k_2 .

```
Program prim9;
type mas=array[1..100,1..100] of integer;
var a:mas; i,j,n,k1,k2,r: integer;
begin
  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 ('k1=,k2='); readln (k1,k2);
    {Меняем значения k1 и k2 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 10 (самостоятельно):

- В массиве размером $n*n$ к элементам четных столбцов прибавить элемент первого столбца соответствующей строки.

Идея решения:

- В данной задаче необходимо проверять четность столбцов.
- Основная часть программы будет выглядеть так:

```
For i:=1 to n do begin  
  For j:=1 to n do
```

- Проверяем номер столбца на четность. В случае четности прибавляем к значениям проверяемого столбца соответствующие значения первого столбца.

```
if j mod 2=0 then a[i,j]:=a[i,j]+a[i,1];  
end;
```

- Далее распечатываем измененный массив.

Иногда массив необходимо заполнять не случайными числами, а по некоторому закону. Рассмотрим такой пример.

Задача 11:

- Заполнить массив A размером $n*m$ следующим образом, например, $n=5$
 $m=5$:

	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».

Решение:

- *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)*
$$a[i,j] = i*m - j + 1.$$
- По этому правилу и составляем процедуру заполнения. Тем более, что проверка на четность производилась в предыдущей задаче (только со столбцами).

```

Program prim11;
  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 imod2=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.

```

Задача 12:

- Латинским квадратом называется массив, в строках и столбцах которого нет одинаковых элементов. Вывести на экран латинский квадрат размером $N \times N$.
- Пример латинского квадрата:

1	2	3	4	5
5	1	2	3	4
4	5	1	2	3
3	4	5	1	2
2	3	4	5	1

Идея решения:

- Заполнить 1-ю строку квадратного массива ($N \times N$) числами от 1 до N .
- Вторая строка массива получается путем циклического сдвига элементов первой строки, и т.д.
- Циклический сдвиг можно реализовать, используя типовой алгоритм вставки-удаления (в зависимости от циклического сдвига).

```
Program prim12;
var a: array[1..10,1..10]
of integer;
n, i, j, x: integer;
begin
  writeln (`razmernost=`);
  readln (n);
  for j:=1 to n do
    a[1,j]:=j;
    {СДВИГ}
  for i:=2 to n do
    begin
      for j:=1 to n do
        a[i,j]:=a[i-1,j];
      x:=a[i,n];
```

```
for j:=n downto 2 do
  a[i,j]:=a[i,j-1];
a[i,1]:=x;
  {ВЫВОД}
  for i:=1 to n do
    begin
      for j:=1 to n do
        write (a[i,j]);
      writeln;
    end;
  readln;
End.
```