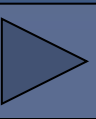


# Массивы

Основные операции над массивами



# Массив – это таблица представленная на языке программирования

## Общая запись одномерных массивов

```
const n=a; {a-число}  
var a:array[1..n] of real; {или другой тип переменных}
```

## Ввод переменного массива

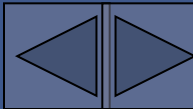
```
{Ввод по одному}  
for i:=1 to n do  
read (a[i]);
```

```
{Заполнение случайными числами}  
randomize;  
for i:=1 to n do  
a[i]:=random(10)
```

## Ввод постоянного массива

```
type a=array[1..3] of string[10]  
const name: mas('a', 'bb', 'ccc');
```

```
const n=5;  
a:array[1..n,1..n] of integer=((1,2,3),(4,5,6));
```



# Виды массивов

```
graph TD; A[Виды массивов] --> B[Одномерный]; A --> C[Двумерный]; A --> D[Трёхмерный];
```

## Одномерный

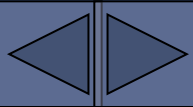
Массив называется одномерным, если состоит из одной строки.

## Двумерный

Массив называется двумерным, если он состоит из нескольких строк.

## Трёхмерный

Массив называется трехмерным, если состоит из нескольких двумерных массивов.



Для работы с массивами в *Turbo Pascal* нужно:

а) определить тип данных

б) задать (описать) массив

в) ввести массив

г) приступить к реализации алгоритма решения  
с последующим выводом результата



# Выберите интересующий вас раздел:

**1. Простые операции с массивами.**

**2. Операции с двумерными массивами.**

**3. Символьные массивы.**



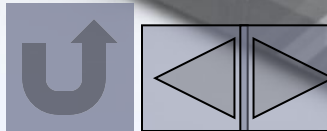
# Простые операции над массивами.

Поиск заданного элемента в массиве.

Подсчет элементов в массивах.

Математические операции над массивом.

Статистические операции над массивом.

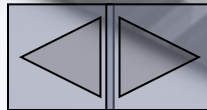


# Простые операции над массивами.

Поиск заданного элемента(ов) в массиве.

```
{Поиск заданного элемента}  
writeln ('Введите нужное число');  
readln(k);  
for i:=1 to n do  
if a[i]=k then writeln ('Число найдено в ', i, ' ячейке');
```

```
90 PRINT "Введите нужный элемент"  
100 INPUT m  
110 FOR i = 1 TO n  
120 IF a(i) = m THEN PRINT "Число найдено! Ячейка "; i  
130 NEXT i
```



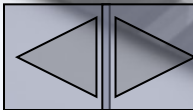


# Простые операции над массивами.

Подсчет элементов в массивах.

```
{Подсчет элементов равных заданному}  
WriteLn ('Введите число');  
ReadLn (k);  
for i:=1 to n do  
for j:=1 to n do  
If a[i,j]=k then s:=s+1;  
Writeln ('Число ',k,' встречается ',s,' раз');
```

```
REM Подсчет элементов больших заданному  
90 PRINT "Введите число"  
100 INPUT m  
110 FOR i = 1 TO n  
120 FOR j = 1 TO n  
130 IF a(i, j) > m THEN LET t = t + 1  
140 NEXT j  
150 NEXT i  
160 PRINT "Количество эл-ов больших числа "; m; t
```



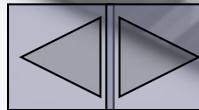


# Простые операции над массивами.

Математические операции над массивами.

```
{Увеличение всех ячеек массива на заданное число}  
WriteLn ('Введите число');  
ReadLn (k);  
for j=1 to n do  
for i=1 to n do  
a[j,i]:=a[j,i]+k;
```

```
REM Деление всех элементов массива на 2  
110 FOR i = 1 TO n  
120 FOR j = 1 TO n  
130 a(i, j) = a(i, j) / 2  
140 NEXT j  
150 NEXT i
```



# Простые операции над массивами.

## Статистические операции над массивами.

```
{Нахождение минимального элемента массива}
min:=a[1];
for i:=2 to n do
if a[i]<min then min:=a[i];
writeln ('Минимальный элемент массива = ',min);
```

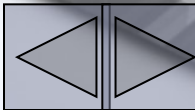
```
{Нахождение максимального элемента массива}
for i:=1 to n do
if a[i]>max then max:=a[i,j];
writeln ('Максимальный элемент массива = ',max);
```

```
{Нахождение среднего значения массива}
for i:=1 to n do
o:=o+a[i];
s:=o/n;
writeln ('Среднее значение массива = ',s:2:2);
```

```
REM Нахождение минимального эл-та
50 LET min = a(1)
60 FOR i = 1 TO n
70 IF a(i) < min THEN LET min = a(i)
80 NEXT i
90 PRINT "минимальный элемент"; min
```

```
REM Нахождение максимального эл-та
60 FOR i = 1 TO n
70 IF a(i) > max THEN LET max = a(i)
80 NEXT i
90 PRINT "максимальный элемент"; max
```

```
REM Нахождение среднего значения
60 FOR i = 1 TO n
70 LET c = c + a(i)
80 NEXT i
90 LET s = c / n
100 PRINT "Среднее значение"; s
```



# Операции с двумерными массивами.

Работа со строками и столбцами.

Периметр матрицы.

Формирование 2-го массива из 1-го.

Перестановки в матрице.

Сортировка.

Повороты матрицы.

Работа с диагоналями матрицы.

Транспонирование матрицы.





# Операции с двумерными массивами.

Работа со строками и столбцами.

```
{Сложение 2-х строк}  
writeln ('Введите 2 строки');  
readln (x,y);  
for j:=1 to m do begin  
  b[j]:=a[x,j]+a[y,j];  
end;
```

```
REM Нахождение разности 2 столбцов модулю  
60 PRINT "Введите номера 2 столбцов"  
70 INPUT x, y  
80 FOR i = 1 TO n  
90 LET b(i) = ABS(a(i, x) - a(i, y))  
100 PRINT b(i)  
110 NEXT i
```



# Операции с двумерными массивами.

Формирование 2-го массива из 1-го.

```
{формирование 2-го массива,  
состоящего из квадратов эл-ов 1-го массива}  
for j=1 to n do  
for i=1 to n do  
b[j,i]:=a[j,i]*a[j,i];
```

```
REM формирование 2-го массива, состоящего из  
REM эл-ов, в 3 раза меньших чем эл-ты 1-го массива  
60 FOR i = 1 TO n  
70 FOR j = 1 TO n  
80 LET b(i, j) = a(i, j) / 3  
90 PRINT b(i, j)  
100 NEXT j, i
```





# Операции с двумерными массивами.

## Сортировка.

```
{Сортировка числового одномерного массива по возрастанию}  
for i:=1 to n-1 do  
  for j:=i+1 to n do  
    if a[j]>a[i] then begin  
      s:=a[i];  
      a[i]:=a[j];  
      a[j]:=s;  
    end;
```

```
REM Сортировка по убыванию  
100 FOR i = 1 TO n - 1  
110 FOR j = i + 1 TO n  
120 IF a(j) < a(i) THEN LET s = a(i): a(i) = a(j): a(j) = s  
130 NEXT j, i
```



# Операции с двумерными массивами.

## Работа с диагоналями квадратной матрицы (матрица – числовой массив).

```
{Нахождение среднего арифметического  
элементов, лежащих ниже главной диагонали}  
for i:=2 to n do  
  for j:=1 to i - 1 do begin  
    s:=s * a(i,j);  
    c:=c + 1;  
  end;  
o:=s/c;  
Writeln (o);
```

```
REM Нахождение произведения элементов над побочной диагональю  
170 p = 1  
180 FOR i = 1 TO n%  
190 FOR j = 1 TO n%  
200 IF i < n% - j + 1 THEN p = p * ar(i, j)  
210 NEXT j, i  
220 PRINT p
```

```
REM Нахождение максимума на побочной диагонали  
100 FOR i = 2 TO n%  
110 IF ar(i, n% - i + 1) > max THEN max = ar(i, n% - i + 1)  
120 NEXT i  
130 PRINT max
```

```
REM Нахождение среднего арифметического элементов,  
лежащих ниже главной диагонали  
240 s = 0  
250 FOR i = 2 TO n%  
260 FOR j = 1 TO i - 1  
270 s = s + ar(i, j): s1 = s1 + 1  
280 NEXT j, i  
300 PRINT s / s1
```

```
{Нахождение минимума на главно  
й диагонали}  
min:=a(1,1);  
for i:=2 to n do  
  if a(i,i) < min then min:=a(i,i);  
Writeln (min);
```

```
REM Нахождение минимума на главной диагонали  
90 min = ar(1, 1)  
100 FOR i = 2 TO n%  
110 IF ar(i, i) < min THEN min = ar(i, i)  
130 NEXT i  
140 PRINT min
```





# Операции с двумерными массивами.

## Периметр матрицы.

```
{Вычисление сумм эл-ов на
периметрах матрицы}
for k:=(n + 1) div 2 to 1 do begin
  sk:=0;
  l:=k;
  r:=n-k+1;
  for i:=1 to r do
    for j:=1 to r do
      sk:=sk+a(i,j);
  d:=sk - s;
  writeln (d);
  s:=sk;
end;
```

```
REM вычисление сумм элементов на периметрах матрицы.
90 FOR k = (n + 1) \ 2 TO 1 STEP -1
100 sk = 0: l = k: r = n - k + 1
110 FOR i = 1 TO r
120 FOR j = 1 TO r
130 sk = sk + a(i, j)
140 NEXT j, i
150 PRINT sk - s;
160 s = sk
170 NEXT k
```



# Операции с двумерными массивами.

## Перестановки в матрице.

```
{Взаимный обмен эл-ов двух строк}  
writeln ('Введите номера 2 строк');  
readln (x,y);  
for j:=1 to m do begin  
  b[j]:=a[x,j];  
  a[x,j]:=a[y,j];  
  a[y,j]:=b[j];  
end;
```

```
REM Взаимный обмен эл-ов 2 столбцов  
50 PRINT "Введите номера 2 столбцов"  
60 INPUT x, y  
70 FOR i = 1 TO n  
80 LET b(i) = a(i, x): a(i, x) = a(i, y): a(i, y) = b(i)  
90 NEXT i
```





# Операции с двумерными массивами.

## Повороты матрицы.

```
{Поворот матрицы на 90 градусов}
for i:=1 to n do begin
  for j:=1 to m do begin
    x[i,j]:=a[m-j+1,i];
    writeln (x[i,j]);
  end;
  writeln;
end;
```

```
{Поворот матрицы на 180 градусов}
for i:=1 to m do begin
  for j:=1 to n do begin
    y[i,j]:=a[m-i+1,n-j+1];
    writeln (y[i,j]);
  end;
  writeln;
end;
```

```
{Поворот матрицы на 270 градусов}
for i:=1 to n do begin
  for j:=1 to m do begin
    z[i,j]:=a[j,n-i+1];
    writeln (z[i,j]);
  end;
  writeln;
end;
```

```
350 REM "Поворот на 90 градусов по часовой стрелке"
360 FOR i = 1 TO n%
370 FOR j = 1 TO m%
380 y(i, j) = ar(m% - j + 1, i)
390 PRINT y(i, j);
400 NEXT j
410 PRINT
420 NEXT i
```

```
220 REM "Поворот на 180 градусов"
230 FOR i = 1 TO m%
240 FOR j = 1 TO n%
250 x(i, j) = ar(m% - i + 1, n% - j + 1)
260 PRINT x(i, j);
270 NEXT j
280 PRINT
285 NEXT i
```

```
420 REM "Поворот на 90 градусов против часовой стрелке"
430 FOR i = 1 TO n%
440 FOR j = 1 TO m%
450 z(i, j) = ar(j, n% - i + 1)
460 PRINT z(i, j);
470 NEXT j
480 PRINT
490 NEXT i
```



# Операции с двумерными массивами.

Транспонирование матрицы.

Транспонирование матрицы – это **симметричный** обмен ячейками, относительно главной диагонали.

```
{Транспонирование матрицы}  
for i:=1 to n-1 do  
  for j:=i+1 to n do begin  
    x:=a[i,j];  
    a[i,j]:=a[j,i];  
    a[j,i]:=x;  
  end;  
end;
```

```
REM Транспонирование  
50 FOR i = 1 TO n - 1  
60 FOR j = i + 1 TO n  
70 LET x = a(i, j): a(i, j) = a(j, i): a(j, i) = x  
80 NEXT j, i
```

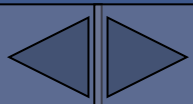


# Операции с символьными массивами.

Поиск.

Отбор по значению.

Сортировка.

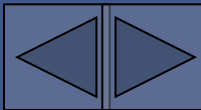




# Операции с символьными массивами.

Поиск.

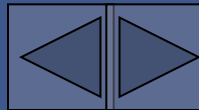
```
{поиск слова в массиве}  
writeln('ввести искомое слово');  
readln(s);  
for i:=1 to 5 do begin  
  for j:=1 to length(a[i]) do  
    if s=copy(a[i],j,length(s)) then n:=n+1;  
end;  
writeln('слово ',s,' встречается ',n,' раз');
```



# Операции с символьными массивами.

Отбор по значению.

```
{Отбор слов начинающихся на опред. букву}  
writeln('ввести нужную букву');  
readln(s);  
  for i:=1 to n do  
    if s=copy(a[i],1,1) then  
      writeln (a[i]);
```



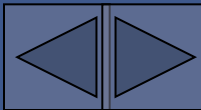


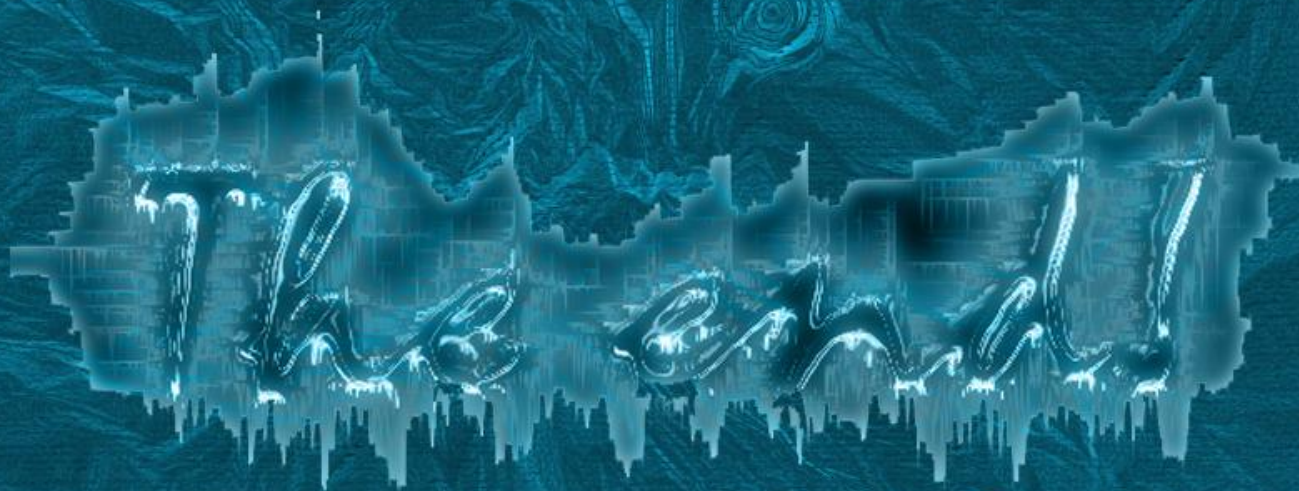
# Операции с символьными массивами.

## Сортировка.

```
{сортировка}
for i:=1 to n-1 do begin
  b:=copy(a[i],1,1);
  for j:=i+1 to n do begin
    b1:=copy(a[j],1,1);
    if b>b1 then begin
      dop:=a[i];
      a[i]:=a[j];
      a[j]:=dop;
      b:=b1;
    end;
  end;
end;
```

```
{сортировка}
for i:=1 to n-1 do begin
  for j:=i+1 to n do begin
    if a[i]>a[j] then begin
      dop:=a[i];
      a[i]:=a[j];
      a[j]:=dop;
    end;
  end;
end;
```





г.Нерчинск (Курилов И.А.)

