

The background of the slide is a blue-tinted photograph of a chalkboard. On the board, there are faint, handwritten mathematical formulas, including what appears to be $\vec{C} = \vec{A} \times \vec{B}$ and $\vec{C} = \vec{A} \cdot \vec{B}$. A piece of white chalk lies on the ledge of the chalkboard in the foreground.

Лекция №6

Одномерные массивы

План лекции

1. Понятие одномерного статического массива
2. Ввод – вывод элементов массива
3. Заполнение массива случайными числами
4. Нахождение суммы элементов
5. Объявление массивов с использованием раздела описания типов
6. Суммирование двух массивов
7. Определение числа элементов, удовлетворяющих некоторому условию
8. Нахождение индексов элементов с заданным свойством
9. Объединение двух массивов в один
10. Поиск максимального и минимального элементов массива
11. Удаление элементов массива
12. Включение элементов в массив
13. Перестановка элементов массива
14. Инвертирование массива
15. Формирование массива из элементов других массивов
16. Циклический сдвиг элементов массива

Одномерный массив

Статический массив – упорядоченная последовательность фиксированного количества переменных одного типа, имеющая общее имя.

Описание массива:

<идентификатор>: array [<диапазон индексов>] of <тип элементов>;

<i>i</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>	<i>10</i>
A[i]	284	345	91	-34	456	3	45	-456	28	23

Пример объявления массива

Пример объявления массива 10-ти целых чисел.

Var

a : array [1..10] of integer;

<i>i</i>	<i>1</i>	<i>2</i>	<i>3</i>	<i>4</i>	<i>5</i>	<i>6</i>	<i>7</i>	<i>8</i>	<i>9</i>	<i>10</i>
A[i]	284	345	91	-34	456	3	45	-456	28	23

индекс

элемент

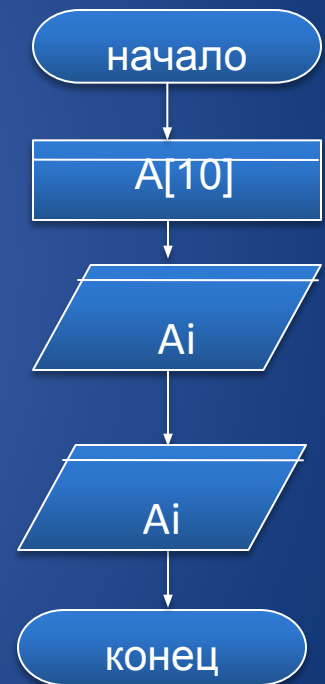
Ввод – вывод элементов массива

Ввод элементов с клавиатуры и вывод элементов.

Так как необходимо ввести определенное число элементов, то алгоритмическая структура программы будет **циклической**. В цикле, управляющей переменной будет являться значение i – индекс элемента массива, для вывода элементов так же будет использован цикл.

Ввод – вывод элементов массива

```
Var
    a : array [1..10] of integer;
    i : integer;
Begin
    writeln ( ' Заполнение элементов целочисленного массива A[10] ' );
    for i:=1 to 10 do
        begin
            write ( 'a[', i , ' ] = ');
            read (a[ i ]);
            end;
        writeln ( 'В памяти компьютера сформирован массив с элементами');
        for i:=1 to 10 do
            write (a[ i ]:6);
        End.
```



Генерация случайных чисел

`random(n)` – функция генерации случайного числа в диапазоне от 0 до $n-1$.

Примеры :

```
x:=random(11);
```

```
y:=random (101)-50;
```

```
z:=random (51)-100;
```

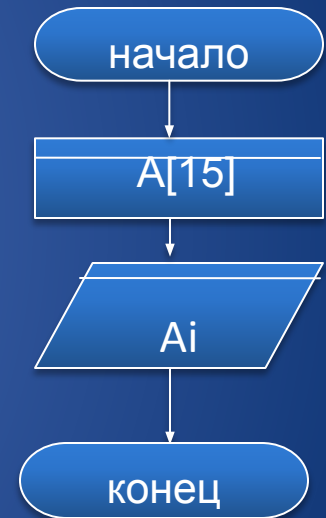
```
k:=random(21)+80;
```

`randomize` – функция позволяющая генерировать случайные числа различными при каждом запуске программы.

Ввод – вывод элементов массива

Генерация элементов массива случайными числами.

```
Var
    a : array [1..15] of integer;
    i : integer;
Begin
    randomize;
    writeln ('Элементы целочисленного массива A[15] сформированные
        случайными числами диапазона от -100 до 100. ');
    for i:=1 to 15 do
        begin
            a[ i ]:=random(201)-100;
            write (a[ i ]:6);
        end
    end
End.
```



Нахождение суммы элементов массива

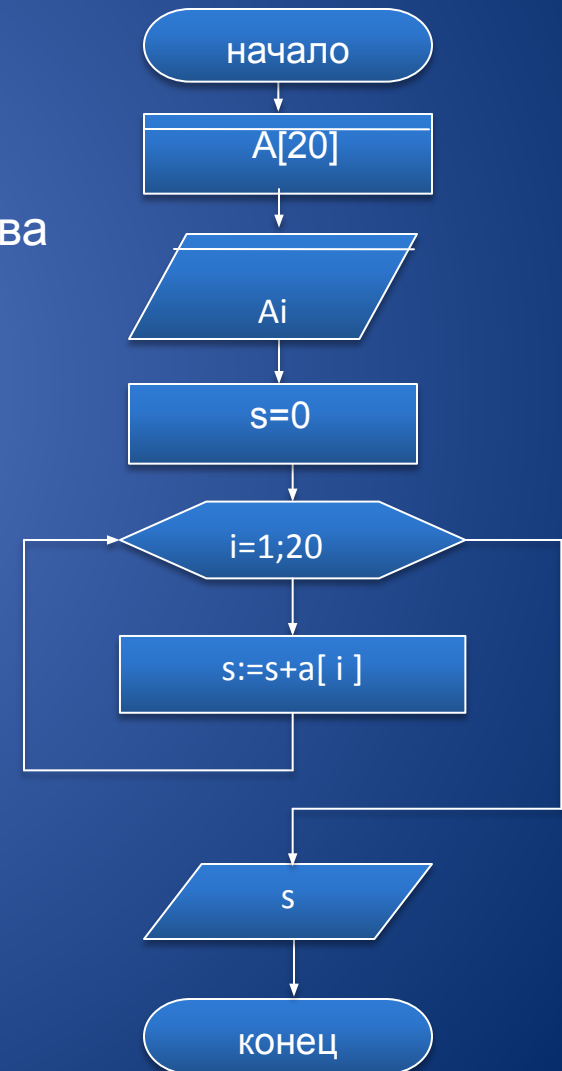
Для получения суммы элементов одномерного статического массива необходимо подготовить ячейку для накопления в нее суммы, предварительно обнулить ее значение. После чего, перебирая в цикле все элементы массива, увеличивать ее значение на величину значения каждого i -го элемента.

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A _i	1	0	-5	4	12	3	30	-2	16	45	11	-1	0	23	0	0	12	26	1	1

S
177

Нахождение суммы элементов массива

```
Var  
    a : array [1..20] of integer;  
    s, i : integer;  
Begin  
    writeln ( ' Заполнение элементов целочисленного массива  
              A[20] ');  
    for i:=1 to 20 do  
        begin  
            write ( 'a[', i , ' ] = ');  
            read (a[ i ] );  
        end;  
    s:=0;  
    for i:=1 to 20 do  
        s:=s+a[ i ];  
    writeln ( 's=',s)  
End.
```



Объявление массивов с использованием раздела описания типов

Пример объявления массива :

Var

a : array [1..50] of real;

b,c : array [1..20] of integer;

...

Аналогичное описание массивов с использованием раздела описания типов:

Type

mas1=array[1..50] of real;

mas2=array[1..20] of integer;

Var

a : mas1;

b,c : mas2;

...

Суммирование двух одномерных массивов

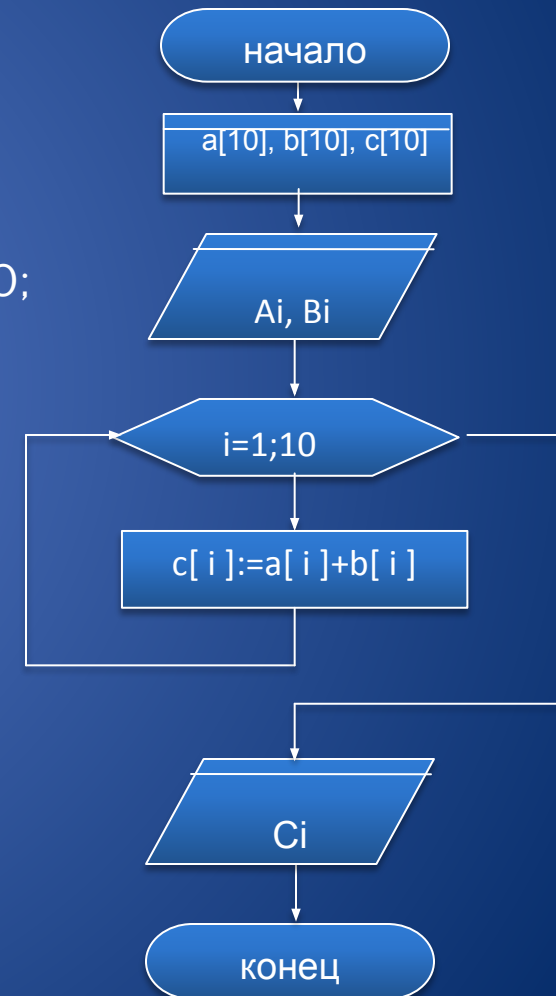
Данный алгоритм подразумевает формирование элементов массива по заданным элементам двух массивов, где каждое очередное значение получаемого массива равно сумме соответствующих элементов заданных массивов по индексом получаемого элемента. Т.е. $c[i]=a[i]+b[i]$.

i	1	2	3	4	5	6	7	8	9	10
Ai	2	10	15	23	7	19	44	3	1	24
Bi	-21	60	64	-54	2	-3	72	-47	2	0
Ci	-19	70	79	-31	9	16	116	-44	3	24

Суммирование двух одномерных массивов

```
Type
    massiv=array[1..10] of integer;
Var
    a , b, c: massiv;
    i : integer;
Begin
    randomize;
    writeln (' Массив A ');
    for i:=1 to 10 do
        begin
            a[ i ]:=random(51);
            write (a[ i ]:5);
        end;
```

```
        writeln (' Массив B ');
        for i:=1 to 10 do
            begin
                b[ i ]:=random(151)-70;
                write (b[ i ]:5);
            end;
        for i:=1 to 10 do
            c[ i ]:=a[ i ]+b[ i ];
            writeln (' Массив C ');
            for i:=1 to 10 do
                write (c[ i ]:6);
            end;
        End.
```



Изменение значений некоторых элементов массива

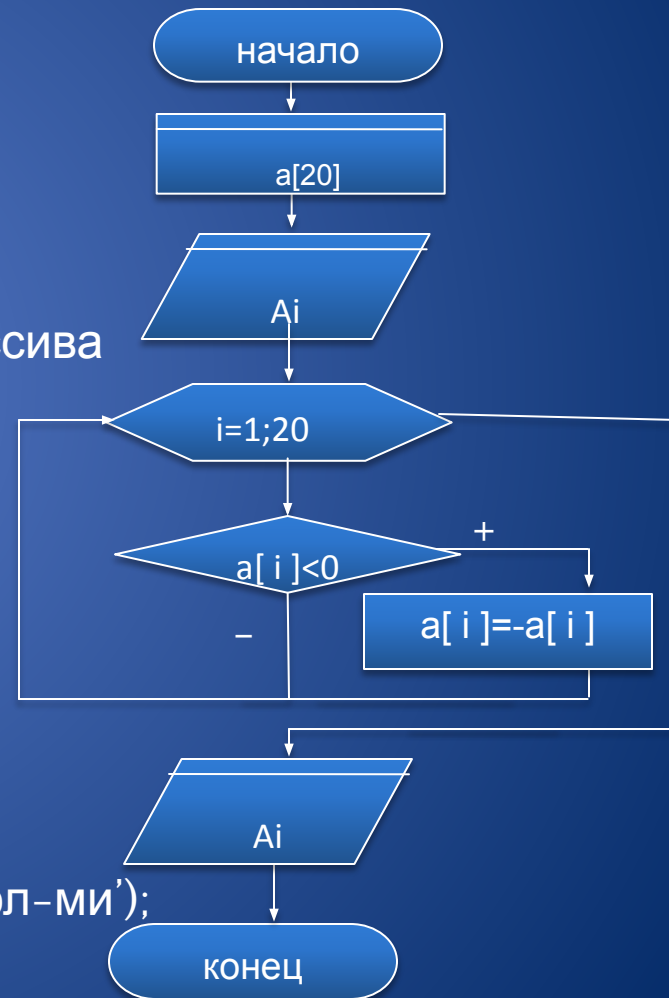
Рассмотрим задачу замены отрицательных элементов на противоположные по знаку.

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A _i	1	0	-5	4	12	3	30	-2	16	45	11	-1	0	23	0	0	12	26	1	1

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A _i	1	0	5	4	12	3	30	2	16	45	11	1	0	23	0	0	12	26	1	1

Изменение значений некоторых элементов массива

```
Const
    n=20;
Var
    a : array [1..n] of integer;
    i : integer;
Begin
    writeln (' Заполнение элементов целочисленного массива
    A[';n;'] ');
    for i:=1 to n do
        begin
            write ('a['; i ; ']=' );
            read (a[ i ]);
        end;
    for i:=1 to n do
        if a[ i ]<0 then
            a[ i ]:=-a[ i ];
    wreteln ('Массив A с замененными отрицательными эл-ми');
    for i:=1 to n do
        write (a[ i ]:5)
    End.
```



Определение числа элементов, удовлетворяющих заданному условию

Для решения такой задачи необходимо задать условие и в цикле перебирая все элементы массива, в случае выполнения данного условия увеличивать значение некоторой переменной k на единицу. До цикла перебора всех элементов, необходимо значение k обнулить. Решим задачу для определения в массиве количества элементов меньших заданного числа T .

T
17

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A_i	19.5	0	-5	41.8	22	31	30.1	-2	16	45	17	-1.5	20	83	17.4	1	12	26	19.1	11

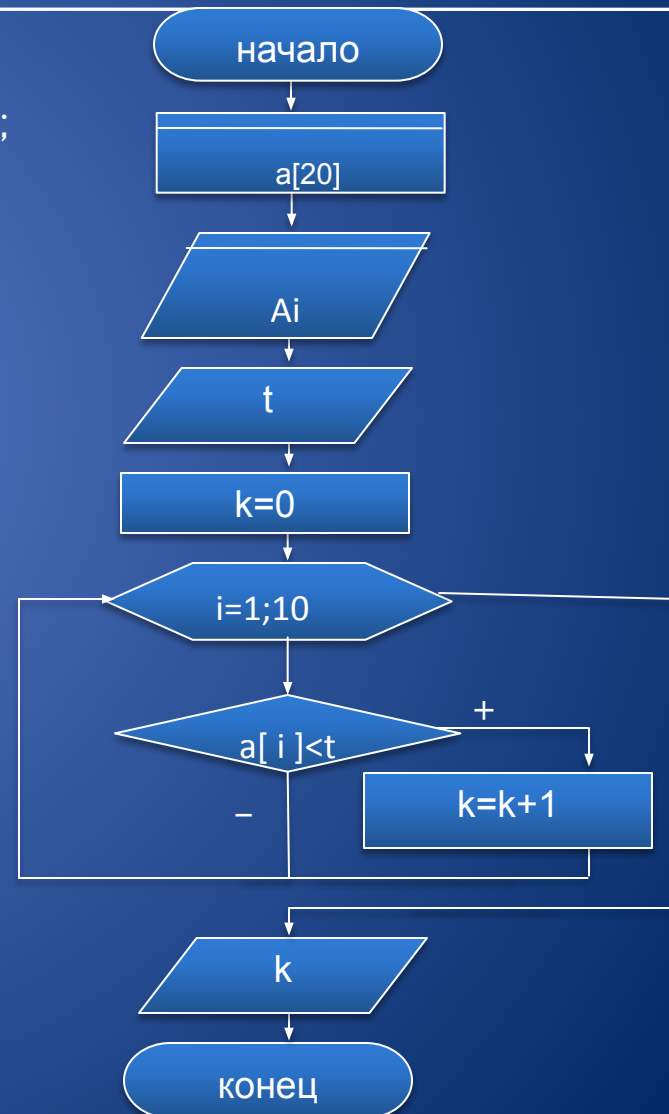
K
8

Определение числа элементов, удовлетворяющих заданному условию

```
Type
    massiv=array[1..20] of real;
Var
    a : massiv;
    t : real;
    k,i : integer;

Begin
    writeln (' Введите элементы
    массива A ');
    for i:=1 to 20 do
        begin
            write ('a[', i , ' ] =');
            read (a[ i ]);
        end;
```

```
writeln (' Введите T');
read (t);
k:=0;
for i:=1 to 20 do
    if a[ i ]<t then
        inc(k);
writeln ('k=',k)
End.
```



Нахождение индексов элементов с заданным свойством

Рассмотрим задачу Нахождения и вывода на экран номеров (индексов) четных элементов.

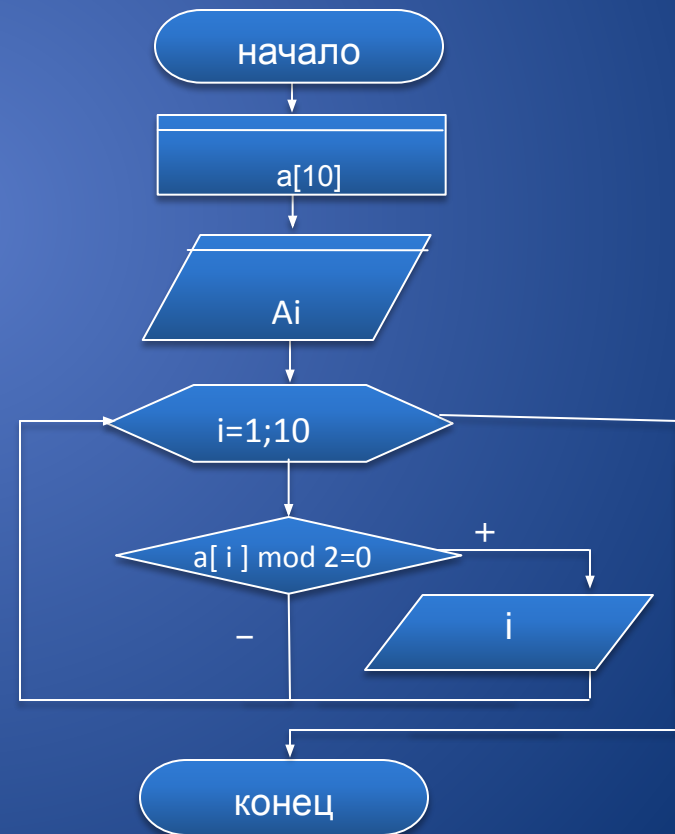
Для решения задачи необходимо просмотреть весь массив, и если просматриваемый элемент является четным, то выводить его индекс.

i	1	2	3	4	5	6	7	8	9	10
Ai	2	10	15	23	7	19	44	3	1	24

1	2	7	10
---	---	---	----

Нахождение индексов элементов с заданным свойством

```
Const
    n=10;
Type
    massiv=array[1..n] of integer;
Var
    a : massiv;
    i: integer;
Begin
    writeln (' Введите элементы
    массива A ');
    for i:=1 to n do
        begin
            write ('a[', i, ']=');
            read (a[ i ]);
            end;
        for i:=1 to n do
            if a[ i ] mod 2=0 then
                write (i:4);
            end;
    End.
```



Объединение двух массивов в один

Задачи по объединению массивов в один имеют различные способы решения. Общим является то, что при формировании элементов объединяемого массива, его значения индексов не будут совпадать со значениями индексов массивов, используемых для объединения. Поэтому основной и главной задачей становится описание закономерности формирования объединенного массива. Например, требуется получить массив С включая элементы массива А и В чередованием : a1;b1;a2;b2.....

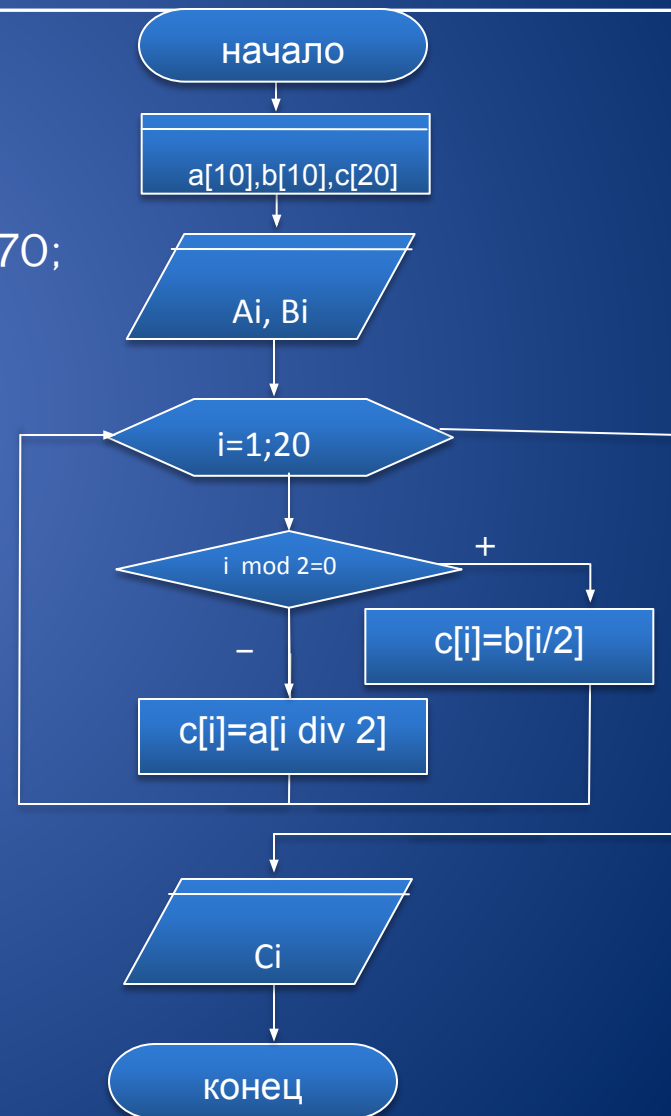
i	1	2	3	4	5	6	7	8	9	10
Ai	2	10	15	23	7	19	44	3	1	24
Bi	-21	60	64	-54	2	-3	72	-47	2	0

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Ci	2	-21	10	60	15	64	23	-54	7	2	19	-3	44	72	3	-47	1	2	24	0

Объединение двух массивов в один

```
Type
    mas1=array[1..10] of integer;
    mas2=array[1..20] of integer;
Var
    a, b : mas1;
    c : mas2;
    i : integer;
Begin
    randomize;
    writeln (' Массив А ');
    for i:=1 to 10 do
        begin
            a[ i ]:=random(51);
            write (a[ i ]:5);
        end;
```

```
writeln (' Массив В ');
for i:=1 to 10 do
    begin
        b[ i ]:=random(151)-70;
        write (b[ i ]:5);
    end;
for i:=1 to 20 do
    if i mod 2=0 then
        c[ i ]:=b[i/2];
    else
        c[ i ]:=a[i div 2];
    writeln (' Массив С ');
    for i:=1 to 20 do
        write (c[ i ]:6)
    End.
```



Поиск минимального и максимального элементов одномерного массива

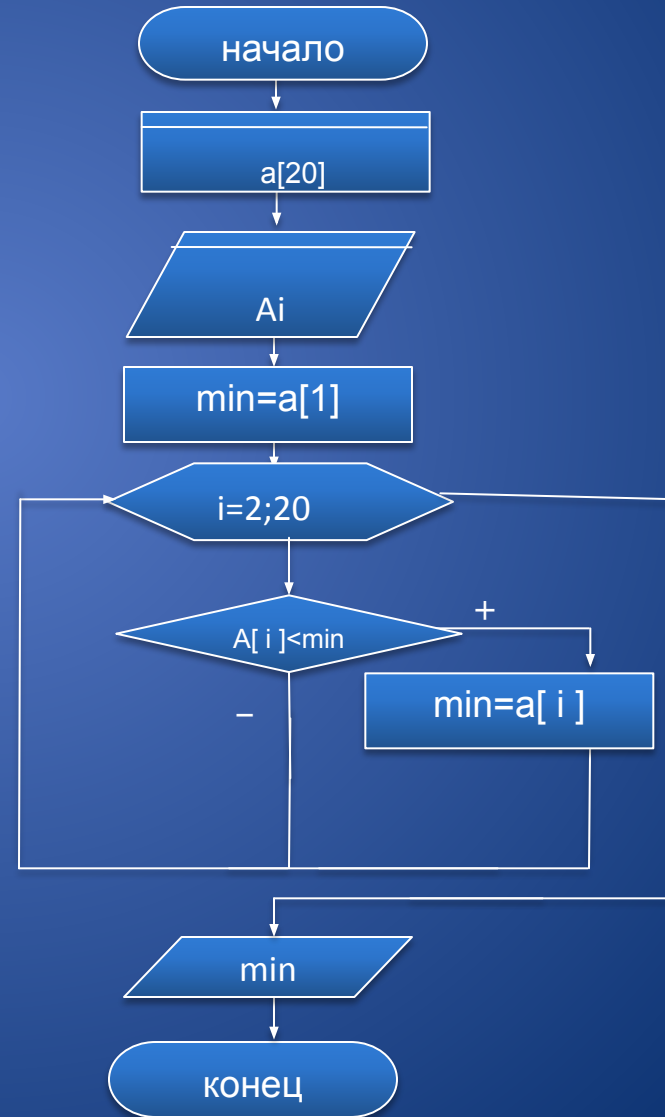
Поиск максимального и минимального элементов массива относится к классическим задачам обработки данных с использованием массива. Суть алгоритма поиска минимального элемента состоит в том, что предположительно за минимальный объявляют первый элемент массива и перебирая все элементы изменяют значение минимального элемента текущим, в том случае, если он оказался меньше минимального на данном этапе. Задача нахождения максимального элемента имеет подобное тривиальное решение.

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A _i	1	0	-5	4	12	3	30	-2	16	45	11	-7	0	23	0	0	12	26	1	1

Min
-7

Поиск минимального элемента одномерного массива

```
Type
mas=array[1..20] of integer;
Var
    a : mas;
    min, i : integer;
Begin
    randomize;
    writeln ( ' Массив ' );
    for i:=1 to 10 do
        begin
            a[ i ]:=random(101)-50;
            write (a[ i ]:6);
        end;
    min:=a[1];
    for i:=2 to 20 do
        if a[ i ] < min then
            min:=a[ i ];
    writeln (min)
End.
```



Удаление элементов массива

Удалить элемент в статическом массиве – невозможно. Поэтому используют перемещение всех элементов, начиная с "удаляемого", записывая на их место следующие $(i+1)$ элементы. Вводят так же переменную, которая обозначает индекс последнего элемента и при каждом шаге удаления элемента ее уменьшают на 1. Рассмотрим задачу на удаление всех отрицательных элементов массива.

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A i	1	0	-5	4	12	3	3.2	-2	16	45	1.1	-7	-1	23	0	0	12	2.6	-1.3	1

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
A i	1	0	4	12	3	3.2	16	45	1.1	23	0	0	12	2.6	1

Удаление элементов массива

```
Type
    mass=array[1..20] of real;
Var
    a: mass;
    i, j, m : integer;
Begin
    randomize;
    writeln ( ' Массив А ');
    for i:=1 to 20 do
        begin
            a[ i ]:=(random(201)-80)/(random(100)+1);
            write (a[ i ]:6:2);
        end;
    m:=20;
    for i:=1 to 20 do
        begin
            if a[ i ]<0 then
```

```
        begin
            for j:=i to 20 do
                a[ j ]=a[ j+1 ];
            dec(m)
        end;
        if a[ i ]<0 then
            dec(i)
        end;
    writeln ( ' Массив А без отрицательных
элементов ');
    for i:=1 to m do
        write (a[ i ]:6:2)
    End.
```

Включение элементов массива

Включить элемент в статический массив так же невозможно. Поэтому изначально размер массива должен быть больше на количество предполагаемых элементов для включения в массив. При включении элемента следует в цикле перебирать элементы от последнего элемента до индекса, куда будет включен элемент и переписывать значения текущего (i -го) элемента на место последующего ($i+1$). Следует так же ввести переменную для хранения индекса последнего элемента, которую при каждом включении увеличивают на 1. Рассмотрим задачу на включение значения T в массив, которое должно располагаться за максимальным элементом массива.

T
5

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A _i	1	0	-5	4	1	3	3	-2	16	45	1.1	-7	-1	23	0	0	12	2.6	-1.3	1

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
A _i	1	0	-5	4	1	3	3	-2	16	45	5	1.1	-7	-1	23	0	0	12	2.6	-1.3	1

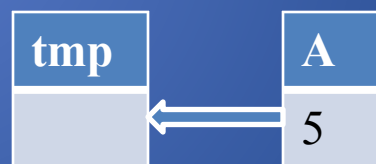
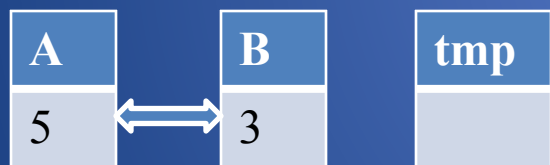
Включение элементов массива

```
Type
    mass=array[1..21] of real;
Var
    a: mass;
    i, j, i_max : integer;
    t, max : real;
Begin
    randomize;
    writeln ( ' Массив А ');
    for i:=1 to 20 do
        begin
            a[ i ]:=(random(201)-80)/(random(100)+1);
            write (a[ i ]:6:2);
        end;
    readln (t);
    max:=a[1];
    i_max:=1;
    for i:=2 to 20 do
        if a[ i ]>max then
```

```
        begin
            max:=a[ i ];
            i_max:=i
        end;
    for j:=21 downto i_max-1 do
        a[ j ]:=a[ j-1 ];
    a[i_max]=t;
    writeln ( ' Массив А с включенным
элементом t ');
    for i:=1 to 21 do
        write (a[ i ]:6:2)
    End.
```

Перестановка элементов массива

Алгоритм перестановки элементов (обмена значениями) прост. Для его выполнения достаточно воспользоваться "временной" переменной, в которую сначала помещают значение первой переменной. Затем в первую переменную заносят значение второй (если не воспользоваться "временной" – значение первой переменной будет потеряно). И сохраненное значение первой переменной во "временной" заносят во вторую переменную. Эту операцию образно можно сравнить с операцией по переливанию двух разных жидкостей из двух пробирок, воспользовавшись третьей – пустой пробиркой.



tmp:=a;



a:=b;



b:=tmp;

Перестановка элементов массива

Рассмотрим задачу обмена максимального и минимального элементов местами.

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
A i	1	0	-5	4	1	3	3	-2	16	45	1.1	-7	-1	23	0	0	12	2.6	-1.3	1	
			max				i_max				min				i_min						
			45				10				-7				12						

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A i	1	0	-5	4	1	3	3	-2	16	-7	1.1	45	-1	23	0	0	12	2.6	-1.3	1

Перестановка элементов массива

Type

```
mass=array[1..20] of real;
```

Var

```
a: mass;
```

```
i, i_max, i_min : integer;
```

```
max, min, tmp : real;
```

Begin

```
randomize;
```

```
writeln (' Массив А ');
```

```
for i:=1 to 20 do
```

```
begin
```

```
a[ i ]:=(random(201)-80)/(random(100)+1);
```

```
write (a[ i ]:6:2);
```

```
end;
```

```
max:=a[1];
```

```
i_max:=1;
```

```
min:=a[1];
```

```
i_min:=1;
```

```
for i:=2 to 20 do
```

```
begin
```

```
if a[ i ]>max then
```

```
begin
```

```
max:=a[ i ];
```

```
i_max:=i
```

```
end;
```

```
if a[ i ]<min then
```

```
begin
```

```
min:=a[ i ];
```

```
i_min:=i
```

```
end
```

```
end;
```

```
tmp:=a[i_min];
```

```
a[i_min]:=a[i_max];
```

```
a[i_max]:=tmp;
```

```
writeln (' Массив А с переставленными  
максимальным и минимальным эл-ми');
```

```
for i:=1 to 20 do
```

```
write (a[ i ]:6:2)
```

```
End.
```

Инвертирование массива

Инвертирование массива – это запись его элементов в обратном порядке. Для решения этой задачи можно воспользоваться другим массивом, в который можно записать элементы из данного массива в обратном порядке. Однако целесообразнее сделать это за наименьшее количество перестановок и не использовать дополнительного массива. Как это сделать? Можно двигаться от первого элемента до середины массива и менять местами первый элемент с последним, второй – с предпоследним и т.д. Получается что в цикле будут обмениваться элемент с i -м индексом с элементом у которого индекс равен $n-i+1$, где n – индекс последнего элемента.

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A _i	1	0	-5	4	1	3	3	-2	16	45	1.1	-7	-1	23	0	0	12	2.6	-1.3	1

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A _i	1	-1.3	2.6	12	0	0	23	-1	-7	1.1	45	16	-2	3	3	1	4	-5	0	1

Инвертирование массива

```
Var
    a: array [1..20] of real;
    i, n : integer;
    tmp : real;

Begin
    randomize;
    writeln ( ' Массив A ');
    n:=20;
    for i:=1 to n do
        begin
            a[ i ]:=(random(201)-80)/(random(100)+1);
            write (a[ i ]:6:2);
        end;
    for i:=1 to n div 2 do
        begin
            tmp:=a[ i ];
            a[ i ]:=a[ n-i+1 ];
            a[ n-i+1 ]:=tmp;
        end;
```

```
writeln ( ' Инвертированный массив A ');
for i:=1 to n do
    write (a[ i ]:6:2)
End.
```

Формирование массива из элементов другого массива

При решении таких задач, сложность их реализации заключается в том, что параллельно с изменением индексов элементов исходного массива необходимо вводить индексы получаемого массива, при том, что индекс получаемого массива должен увеличиваться только в том случае, если элемент включается в новый массив. Размер нового массива следует принять равным исходному, потому что может оказаться так, что все элементы будут включены в новый массив. Рассмотрим задачу с включением в новый массив элементов, у которых значения являются большими заданного значения T .

T
17

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
A_i	19.5	0	-5	41.8	22	31	30.1	-2	16	45	17	-1.5	20	83	17.4	1	12	26	19.1	11

i	1	2	3	4	5	6	7	8	9	10	11
B_i	19.5	41.8	22	31	30.1	45	20	83	17.4	26	19.1

Формирование массива из элементов другого массива

```
Var
    a,b: array [1..20] of real;
    i , n : integer;
    t : real;

Begin
    randomize;
    writeln ( ' Массив А ');
    n:=20;
    for i:=1 to n do
        begin
            a[ i ]:=(random(201)-80)/(random(100)+1);
            write (a[ i ]:6:2);
        end;
    readln (t);
    j:=0;
    for i:=1 to n do
        if a[ i ]>t then
            begin
                inc(j);
                b[ j ]:=a[ i ]
```

```
            end;
        if j=0 then
            writeln ( 'Массив В не содержит
                элементов')
        else
            begin
                writeln ( 'Массив В ');
                for i:=1 to j do
                    write (b[ i ]:6:2)
                end
            end
        End.
```


Циклический сдвиг элементов массива

Под циклическим сдвигом понимают перемещение элементов "по цепочке". Т. е. при сдвиге элементов вправо элемент находящийся на последнем месте в массиве становится на место первого с последующим сдвигом всех остальных элементов. И наоборот, при циклическом сдвиге влево – первый элемент переходит на место последнего с соответствующим перемещением всех элементов массива. При реализации таких алгоритмов следует помнить, что обход при перемещении циклическим сдвигом вправо начинают от последнего элемента, а при сдвиге влево с первого элемента. Для хранения данных в процессе перемещения элементов можно воспользоваться дополнительным массивом или же просто – одной "временной" переменной.

k
3

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Ai	19.5	0	-5	41.8	22	31	30.1	-2	16	45	17	-1.5	20	83	17.4	1	12	26	19.1	11

i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Ai	41.8	22	31	30.1	-2	16	45	17	-1.5	20	83	17.4	1	12	26	19.1	11	19.5	0	-5

Циклический сдвиг элементов массива влево на k позиций

```
Const
    n=20;
Var
    a: array [1..n] of real;
    i, j, k: integer;
    tmp: real;
Begin
    randomize;
    writeln (' Массив А ');
    for i:=1 to n do
        begin
            a[ i ]:=(random(201)-80)/(random(100)+1);
            write (a[ i ]:6:2);
        end;
    readln (k);
    for i:=1 to k do
        begin
            tmp:=a[1];
```

```
        for j:=1 to n-1 do
            a[ j ]:=a[ j+1 ];
        a[n]:=tmp;
    end;
    writeln ('Массив А сдвинут циклически на k позиций влево');
    for i:=1 to n do
        write (a[ i ]:6:2)
    End.
```

