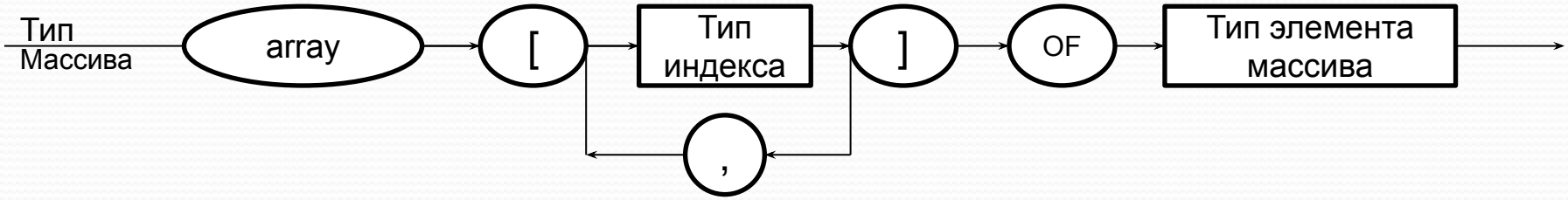
МАССИВЫ

Презентация выполнена
Студентами группы МП 22- 11
Бойцовой Анной и
Набатниковым Александром

Массивы

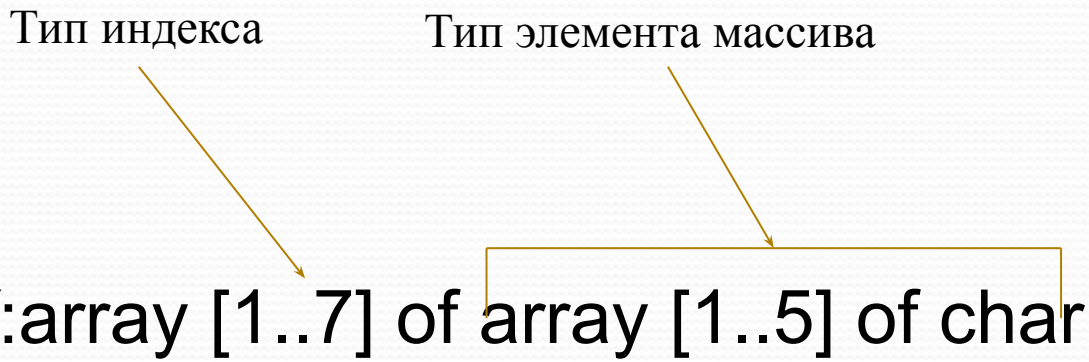
Упорядоченная ограниченная последовательность однотипных элементов, имеющих общее имя называется ***массивом***. Любой массив, использующийся в программе, должен быть объявлен с помощью типа массива.




```
Var : z:array [1..20] of real;
      x:array [1..4,1..3] of integer;
```

Тип индекса - это скалярный тип кроме REAL и неограниченного integer , т.е. может быть использован лишь такой тип , значение которого образует ограниченное пронумерованное множество (Char, boolean, ограниченный, целый)

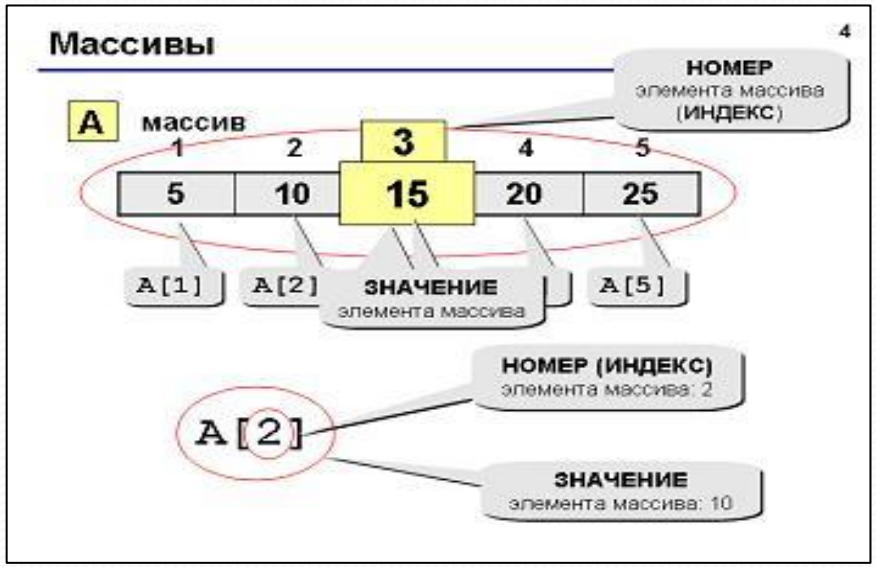
Тип элемента массива- может быть любой, в том числе и тип массива.



Размер массива задаётся при объявлении, и не изменяется в процессе выполнения программы.

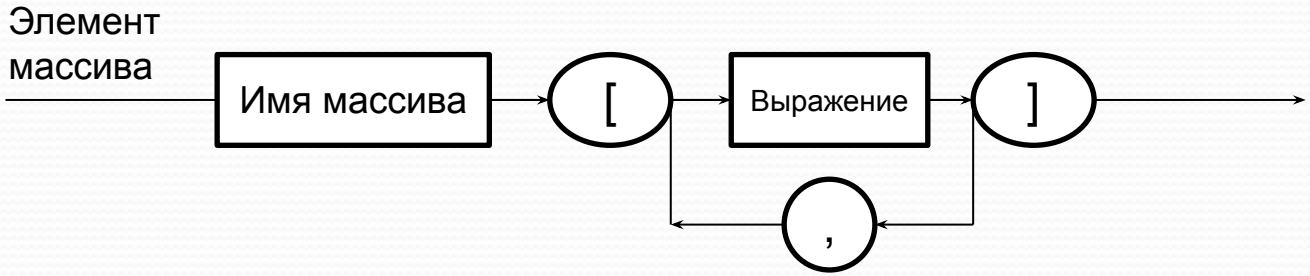
Компилятор по описанию массива резервирует место в памяти, размещая элементы в подряд идущих ячейках , в зависимости от типа элементов и их количества.

(Каждый элемент массива определяется именем массива и индексом)



Обращение к элементам массива

Для того, чтобы обратится к элементу массива необходимо указать имя и индекс массива, который определяет положение элемента в массиве (переменная с индексом).



Индекс- Выражение того же типа, что и тип индекса при объявлении массива.

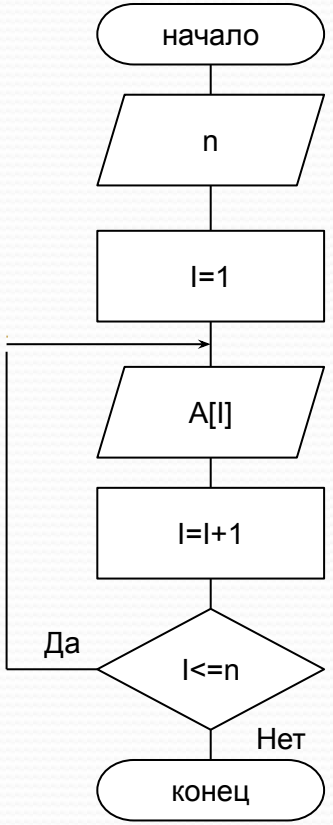
Количество элементов, к которым можно обратится в программе должно соответствовать размеренности массива**.**

Над элементами массива можно производить те же операции, которые допустимы над данными **соответствующего базового типа.**

Вводить и выводить массив можно только по элементам, организовав цикл.

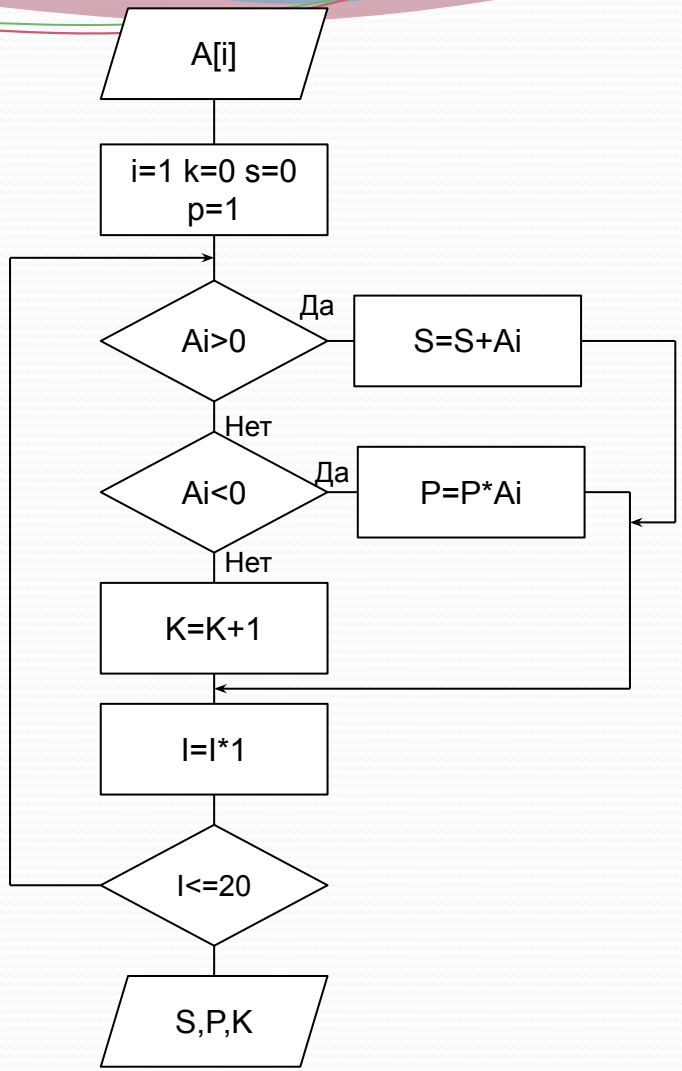
Например

```
Var a:array[1..50] of real;
…
Begin
 Writeln('Введите количество элементов массива');
 Read(n);
  For i:=1 to n do
   begin
    read(A[i]);{Запись элементов массива}
    Writeln(A[i]);{Вывод элементов массива}
   end;
 End.
```

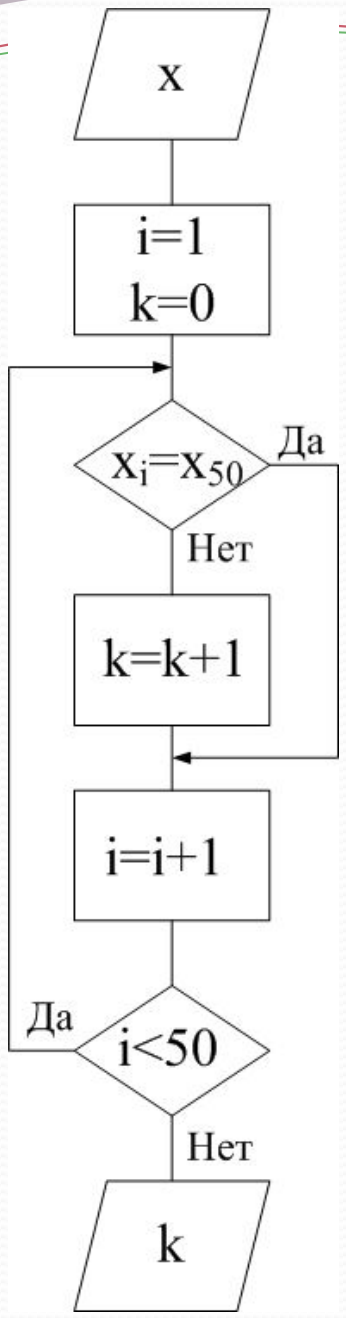


Составить программу подсчета в векторе A(20) количества нулевых элементов, сумму положительных элементов и произведение отрицательных элементов.

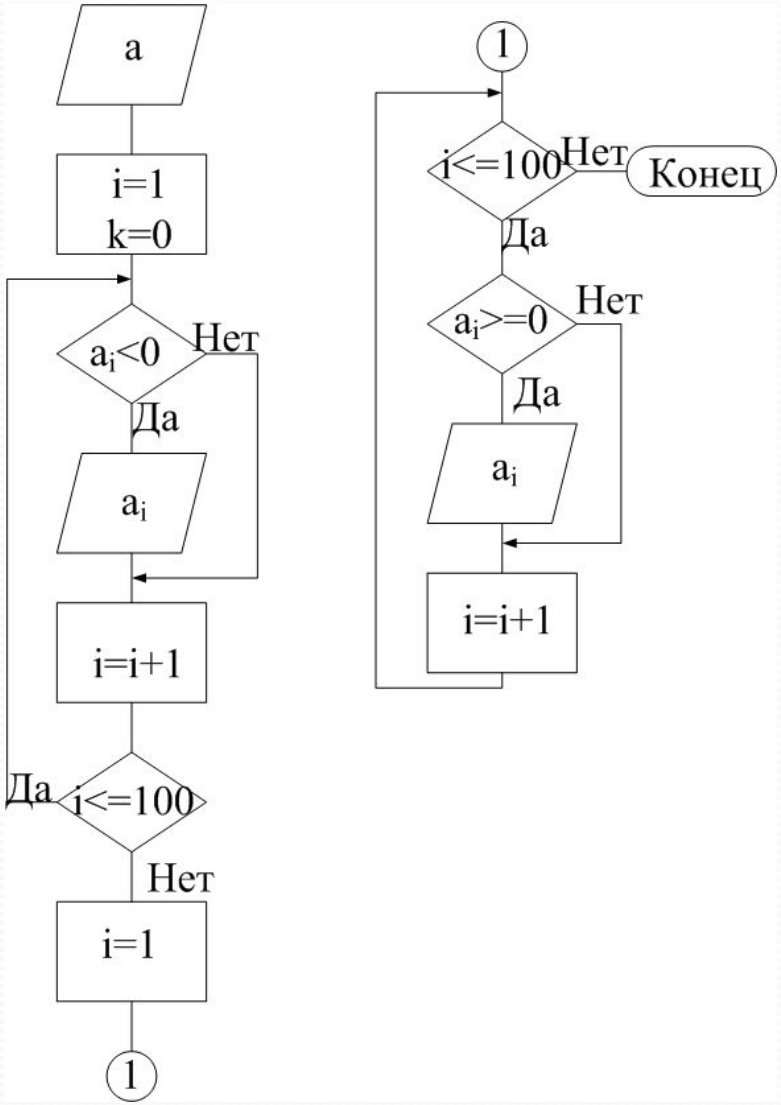
```
Program mas1;
var
  I, K: integer;
  S, P: real;
  A: array[1..20]of real;
begin
  Writeln('Введите элементы массива');
  for i := 1 to 20 do
    Read(A[i]);
  K := 0; s := 0;p := 0;
  for i := 1 to 20 do
   begin
    if A[i] > 0 then s := s + A[i]
      Else
       if A[i] < 0 then p := p * A[i]
      Else K := K + 1;
   end;
  Writeln('s=', s:4:2);
  Writeln('p=', p:4:2);
  Writeln('k=', k:2);
end.
```



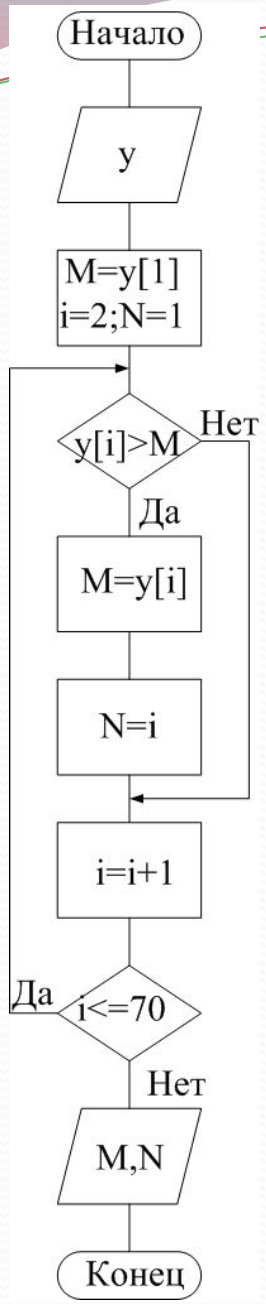
Задание




```
Program MAS2 (input,output);
  var  i, k: integer;
       x: array [1..50] of real;
  Begin
    for i :=1 to 50 do
      read(x[i]);
       k:=0;
    for i:=1 to 50 do
      if not (x[i]=x[50])  then
        k:=k+1;
       writeln('k= ');
  End.
```




```
Program MAS3 (input,output);
  var  i, : integer;
       a: array [1..100] of real;
  Begin
    for i :=1 to 100 do
      if  a[i]<0  then
        writeln (a[i]);
    for i:=1 to 100 do
      if  a[i]>=0  then
        writeln (a[i]);
  End.
```



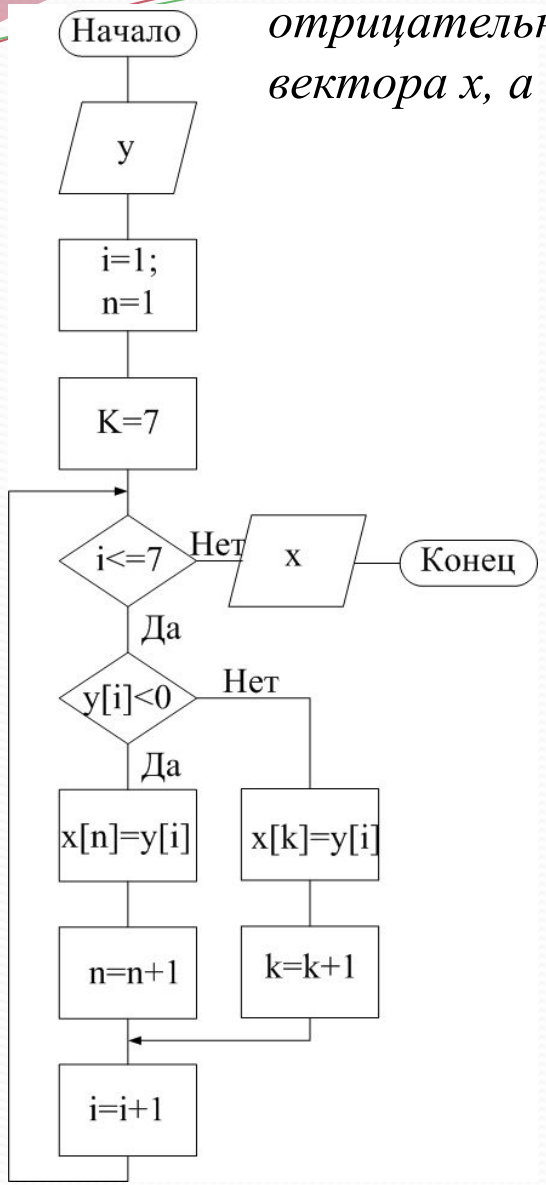
Задача.

Для массива y(70) найти max элемент массива и его номер.

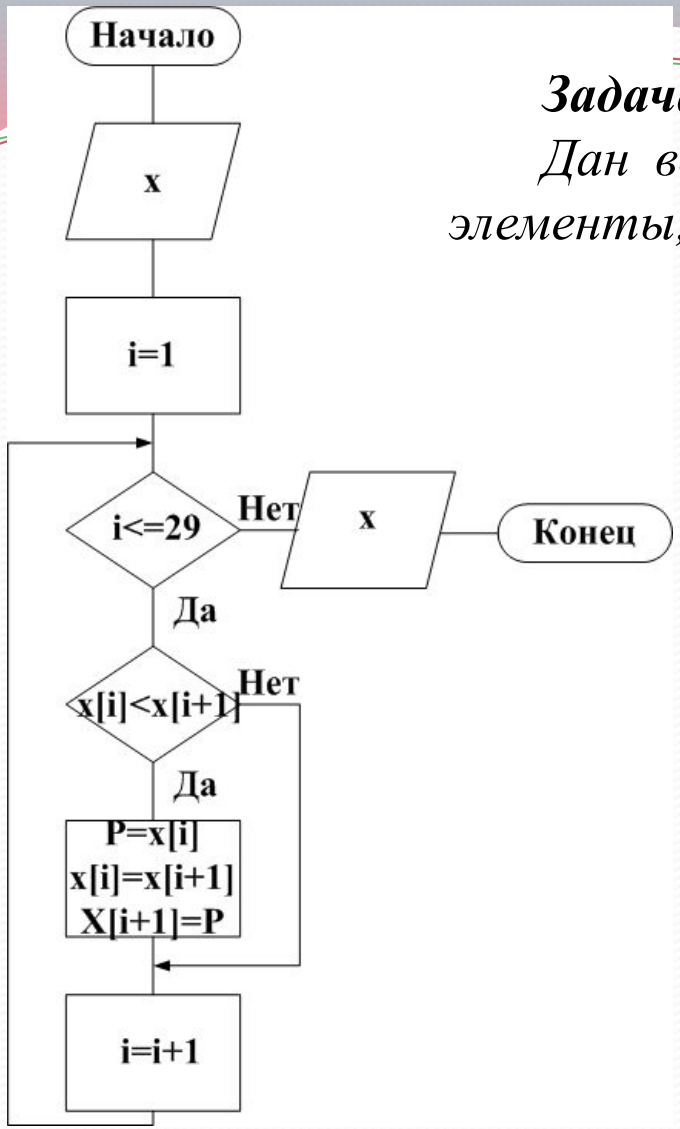
```
Program MAS4 (input,output);
  var  i, N : integer;
       a: array [1..70] of real;
  Begin
    for i :=1 to 70 do
        read(y[i]);
         N:=1;
         M:=y[1];
      if  y[i]>M  then
          begin
            M:=y[i];
            N:=i;
          end;
        writeln (‘M=‘,M);
        writeln (‘N=‘,N);
  End.
```

Задача.

Сформировать из вектора y(7) вектор x(7), так, чтобы отрицательные элементы вектора y располагались в начале вектора x, а остальные элементы - в конце.




```
Program MAS5 (input,output);
 var  i, N,k : integer;
      x,y: array [7] of real;
 Begin
   for i :=1 to 7 do
       read(y[i]);
        N:=1; k:=7;
   for i :=1 to 7 do
    if  y[i]<0  then
        begin
          x:=y[i];
          N:=N+1;
        end
       else
        begin
          x[k]:=y[i];
          k:=k-1;
        end;
       write('x=[');
        for i :=1 to 7 do
         write(x[i],' ');
         writeln(' ]');
End.
```



Задача.

Дан вектор x(30). Необходимо поменять местами элементы, если левый элемент меньше правого.

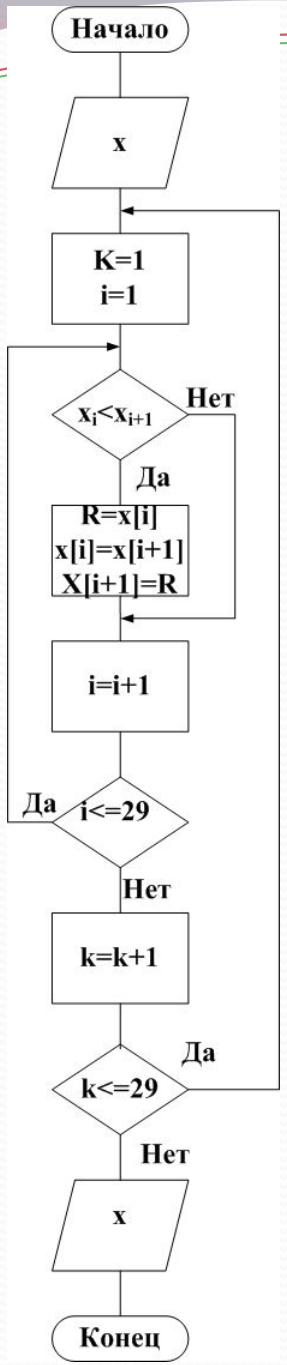
```
Program MAS6 (input,output);
 var  i: integer; P:real;
      x: array [1..30] of real;
 Begin
   for i :=1 to 30 do
       read(x[i]);
  for i :=1 to 29 do
    if  x[i]<x[i+1]  then
        begin
          P:=x[i]; x[i]:=x[i+1];
          x[i+1]:=P;
        end;
     write('x=[');
        for i :=1 to 30 do
         write(x[i],' ');
         writeln(' ]');
 End.
```

С помощью этого алгоритма минимальный элемент вектора окажется на последнем месте.

Упорядоченный массив.

Составить программу по упорядочения вектора в порядке убывания.

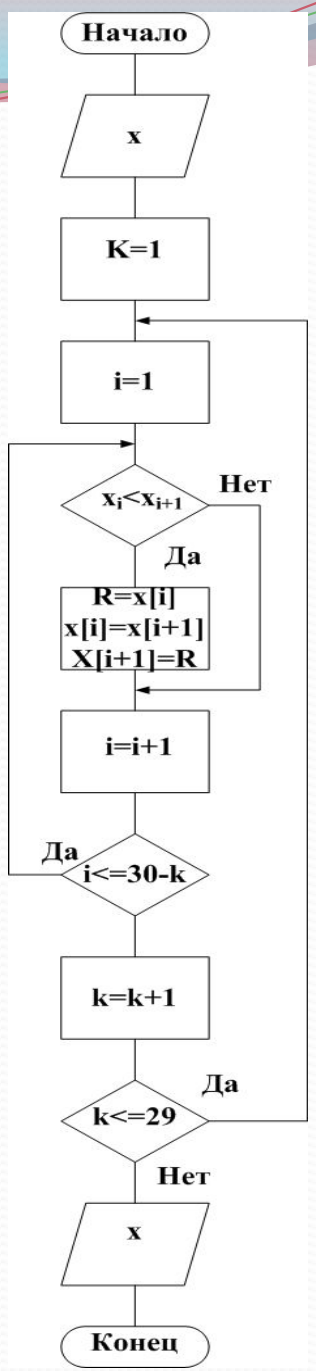
Для упорядочения вектора необходимо организовать сложный цикл, где внутренний цикл- это предыдущая задача(просмотр 2-х соседних элементов и, перестановка их значений, если левый элемент меньше правого), а внешний цикл - это организация просмотра вектора(n-1) раз, если n кол-во элементов вектора.




```
Program MAS67 ;
  var  i,j: integer;
         R:real;
         x: array [1..30] of real;
  Begin
    for i :=1 to 30 do
        read(x[i]);
     for i :=1 to 29 do
         if  x[i]<x[i+1]  then
           begin
             R:=x[i];
             x[i]:=x[i+1];
             x[i+1]:=R;
           end;
            write('x=[');
             for i :=1 to 30 do
              write(x[i],' ');
  End.
```

Внутренний цикл этого алгоритма производит лишние проверки т.к. минимальный элемент оказывается на последнем месте уже за 1-ый просмотр.

k	i
1	29
2	28
3	27



Если за время очередного просмотра произошла хотя бы одна перестановка, то необходимо вектор еще раз просмотреть. Для этого необходимо ввести переменную призрак(«флаг»)

$$P=\begin{cases} 0 - \text{перестановки не было} \\ 1 - \text{перестановка была} \end{cases}$$

Просмотр заканчивается когда не было ни одной перестановки.