Сортировка массивов Сортировка *ShakerSort*

```
Var
    I,j,x,n:integer;
 a:arrray[0...50] of integer;
begin
for i:=1 to n do read(a[i]);
    for i:=2 to n do
      for j:=n downto i do
        if a[j-1]>a[j] then
        begin
           x:=a[j-1];
           A[j-1]:=a[j];
           A[j]:=x;
         End;
 writeIn('Результат:');
For i:=1 to n do write (a[i]);
End.
```

Запоминать, были перестановки в процессе некоторого прохода.

Запоминать не только сам факт, что обмен имел место, но и положение (индекс) последнего обмена.

- Чередовать направление последовательных просмотров.
- Левая граница = Номер первого элемента
- Правая граница = Номер последнего элемента
- Пока Левая граница < Правой границы делать</p>
- Прямой проход "Пузырька" от Левой границы до Правой- I
 - 🜓 🛾 Правая граница = Правая граница І
- Обратный проход "Пузырька" от Правой границы до Левой+ І
 - Левая граница = Левая граница + I

исходный массив:

33 7 15 11 0 4 25 -1

Первый шаг работы алгоритма:

33 7 15 11 0 4 25 -1 => 7 33 15 11 0 4 25 -1.
7 33 15 11 0 4 25 -1 => 7 15 33 11 0 4 25 -1.
7 15 33 11 0 4 25 -1 => 7 15 11 33 0 4 25 -1
7 15 11 33 0 4 25 -1 => 7 15 11 0 33 4 25 -1
7 15 11 0 4 33 25 -1 => 7 15 11 0 4 25 33 -1
7 15 11 0 4 25 33 -1 => 7 15 11 0 4 25 -1 33

Затем меняем направление просмотра и продолжаем процесс

Затем меняем направление просмотра и продолжаем процесс:

Второй шаг работы алгоритма

-1 0 7 11 4 15 25 33

Второй шаг работы алгоритма

-1 0 4 7 11 15 25 33

```
k:= 25; {Индекс последнего изменения}
s:= I; {Первый элемент массива}
е:= 25; {Последний элемент массива}
while e > s do
begin
    for i:= e downto s+1 do if A[i] < A[i-1] then
    begin
         tmp := A[i];
         A[i] := A[i-1];
         A[i-1] := tmp;
          k := i; {запоминание индекса последней перестановки}
     end;
     s:=k;
    for i:= s to e-1 do if A[i]>A[i+1] then
    begin
          tmp := A[i];
         A[i] := A[i+1];
         A[i+1] := tmp;
          k := i:
     end;
     e:=k;
end;
```

