

# **Сортировка элементов в массиве**

Сортировка представляет собой процесс упорядочения элементов в массиве в порядке возрастания или убывания их значений. Например, массив  $X$  из  $n$  элементов будет отсортирован в порядке возрастания значений его элементов, если

$$X[1] \leq X[2] \leq \dots \leq X[n],$$

и в порядке убывания, если

$$X[1] \geq X[2] \geq \dots \geq X[n].$$

# **Сортировка методом прямого обмена ( «пузырька» )**

Сортировка методом «пузырька» основана на выполнении в цикле операций сравнения и при необходимости обмена соседних элементов

Сравним первый элемент массива со вторым, если первый окажется больше второго, то поменяем их местами. Затем сравним второй с третьим, и если второй окажется больше третьего, то поменяем и их. Далее сравниваем третий и т.д.

Если продолжить сравнение соседних элементов: сравнить четвертый с пятым, пятый с шестым и т. д. до сравнения  $n - 1$ -го и  $n$ -го элементов, то в результате этих действий самый большой элемент станет на последнее ( $n$ -е) место. **Это значит, что за один проход на свое место гарантированно встает один элемент.** Теперь повторим данный алгоритм сначала с 1-го до  $n - 1$  элемента (последний элемент, рассматривать не будем, так как он уже занял свое место). После проведения данной операции самый большой элемент оставшейся части массива станет на свое  $m$  ( $n - 1$ -е) место. Так повторяем до тех пор, пока не упорядочим весь массив.

Исходный массив	2	3	5	8	6
1 проход	3	5	8	6	2
2 проход	5	8	6	3	2
3 проход	8	6	5	3	2
4 проход	8	6	5	3	2

```
program upor_massiv ;
var i , j , n : byte ;
X: array [ 1 .. 100 ] of real ;
b : real ;
BEGIN
readln ( n ) ;
for i :=1 to n do
readln ( X[ i ] ) ;

for j:=1 to n-1 do
for i:=1 to n-j do
if X[i]>X[i+1] then {Если текущий элемент больше следующего, то}
begin {поменять их местами.}
b:= X[i]; {Сохранить значение текущего элемента.}
X[i]:=X[i+1]; {Заменить текущий элемент следующим.}
X[i+1]:=b; {Заменить следующий элемент переменной b.}
end;
writeln ( 'упорядоченный массив ' ) ;
for i :=1 to n do
write ( X[ i ] : 5 : 2 , ' _ ' ) ;
writeln ;
END.
```

# **Сортировка методом прямого выбора**



Суть метода заключается в том, что находится минимальный (максимальный) элемент массива и записывается в первую ячейку массива, содержимое которой записывается на место найденного минимального (максимального) элемента. После чего находится минимальный (максимальный) элемент массива, начиная со второго элемента, он записывается во вторую ячейку массива, содержимое которой записывается на место найденного минимального (максимального) элемента. Таким образом, постепенно выстраивается упорядоченный массив. Из алгоритма видно, что аналогично предыдущему методу, за один проход на свое место гарантированно встает один элемент, но в отличие от метода «пузырька», в данном случае массив будет заполняться сначала

Исходный массив	8	6	4	9	2
1 проход	2	6	4	9	8
2 проход	2	4	6	9	8
3 проход	2	4	6	9	8
4 проход	2	4	6	8	9

```
const N=10; {Количество элементов массива}
var a: array[1..N] of integer; {массив}
i,j: integer; {счётчики для цикла}
c: integer; {Переменная для промежуточного хранения}
c2: integer; {Переменная для промежуточного
хранения}
.....
for i:=1 to N-1 do begin {цикл по первому
обрабатываемому элементу массива}
c2:=i; {индекс предполагаемого минимального
элемента}
for j:=i+1 to N do {поиск минимального элемента} if
a[c2]>a[j] then
c2:=j; {если в c2 индекс не минимального элемента, то в
c2 записывается индекс меньшего элемента}
c:=a[i];
a[i]:=a[c2];
a[c2]:=c; {Меняем местами элемент массива}
end;
```