



# Массивы

**Массив** – это группа переменных одного типа, расположенных в памяти рядом (в соседних ячейках) и имеющих общее имя. Каждая ячейка в массиве имеет уникальный номер.

Номер элемента (индекс)	0	1	2	3	4
Значение Элемента	5	10	15	20	25
Обращение к элементу массива	A[0]	A[1]	A[2]	A[3]	A[4]

**A[1]**

НОМЕР (ИНДЕКС)  
элемента массива: 1

ЗНАЧЕНИЕ  
элемента массива: 10

# Запись элемента массива

**Массивы в SB** динамические, т.е. объявлять количество элементов не нужно

- **For i=0 To N-1** // параметрический цикл по i прямой проход
- `ms[i]=i` // считывание i-го элемента
- `mss[i]=ms[i]` // создание копии массива
- **EndFor**

# Проход массива, обратный

**Массивы в SB динамические, т.е. объявлять количество элементов не нужно**

- **For i=N-1 To 0 step -1** // параметрический цикл обратный проход
- LCD.Text(1, 10, 20, 2, ms[i]) // вывод элемента на экран
- **EndFor**

**Как вывести все элементы массива в одном цикле в одну строчку/столбец?**

# Чтение элемента массива из файла

**Массивы в SB** динамические, т.е. объявлять количество элементов не нужно

```
ff1=EV3File.OpenRead("1.in")  
ff2=EV3File.OpenWrite("1.out")  
EV3File.ReadLine(ff1,light2)  
EV3File.ReadNumberArray (ff1, size)
```

**ms[0]=0 // инициализация массива**

**For i=0 To N-1 // параметрический цикл прямой проход**

**ms[i]=EV3File.ReadLine(ff1,i) // считывание i-го  
элемента**

**mss[i]=ms[i] // создание копии массива**

**EndFor**

# Операции с массивами

1. Посчитать сумму элементов массива
2. Посчитать среднее арифметическое элементов массива
3. Посчитать количество элементов  $>/<$  нужного
4. Найти нужный элемент массива и вывести его индекс

# Поиск максимального

## элемента

`nMax=0`

`for i=1 to N-1`

`if A[i] > A[nMax] then`

`nMax=i`

`EndIf`

`EndFor`

`LCD.Text(1, 10, 60, 2, nMax)`

`LCD.Text(1, 10, 80, 2, A[nMax])`

**Отобразить нужные элементы по условию в другой массив**

**Вывести на экран номера элементов массива, не совпадающих с элементами 2-го массива**

# Сортировка массива

**Сортировка** – это расстановка элементов массива в заданном порядке.

## **Алгоритмы:**

**простые и понятные**

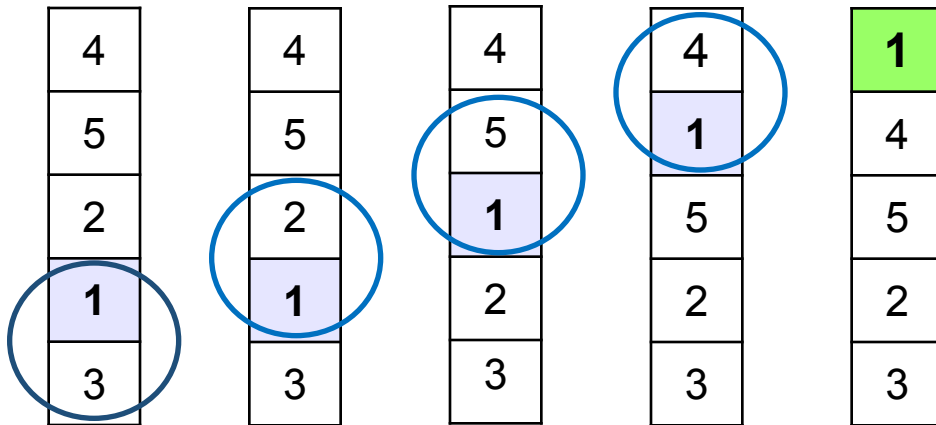
- 1. метод пузырька**
- 2. метод выбора**



# Сортировка массива, пузырек

Для массивов – **самый маленький**  
элемент перемещается вверх.

1-й проход:

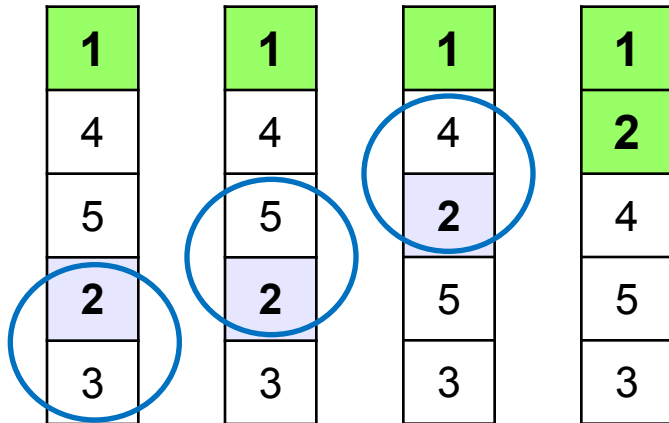


- сравниваем **два соседних** элемента; если они стоят «неправильно», меняем их местами
- за 1 проход по массиву **один** элемент (самый маленький) становится на свое место

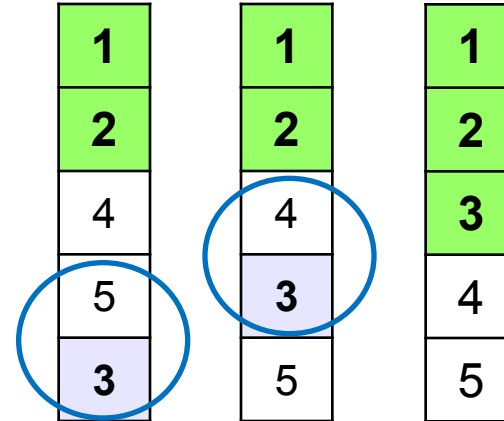
# Сортировка массива,

## пузырек

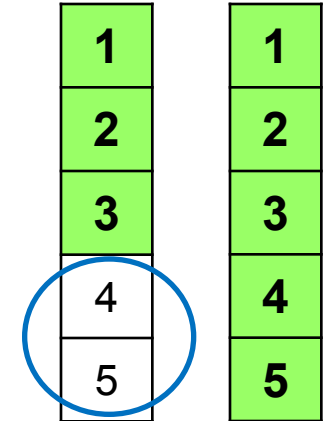
2-й проход:



3-й проход:



4-й проход:



Для сортировки массива из  $N$  элементов нужен  $N-1$  проход (достаточно поставить на свои места  $N-1$  элементов).

# Сортировка массива

Sub Sort

**For k=0 To N-1**

**For j=0 To (N-1-i)-1**

**If mss[k]>mss[j] Then**

temp=mss[k]

mss[k]=mss[j]

mss[j]=temp

**EndIf**

**EndFor**

**EndFor**

**EndSub**

# Сдвиг элементов массива

**Sub Sdvig**

**Temp=ms[0]**

**For i=0 To (N-1)-1**

**ms[i]=ms[i+1]**

**EndFor**

**Ms[N-1]=Temp**

**EndSub**

# Сдвиг элементов массива

**Sub Sdvig**

**Temp=ms[0]**

**For i=0 To (N-1)-1**

**ms[i]=ms[i+1]**

**EndFor**

**Ms[N-1]=Temp**

**EndSub**