



# МЕТОДЫ СОРТИРОВКИ МАССИВОВ

---

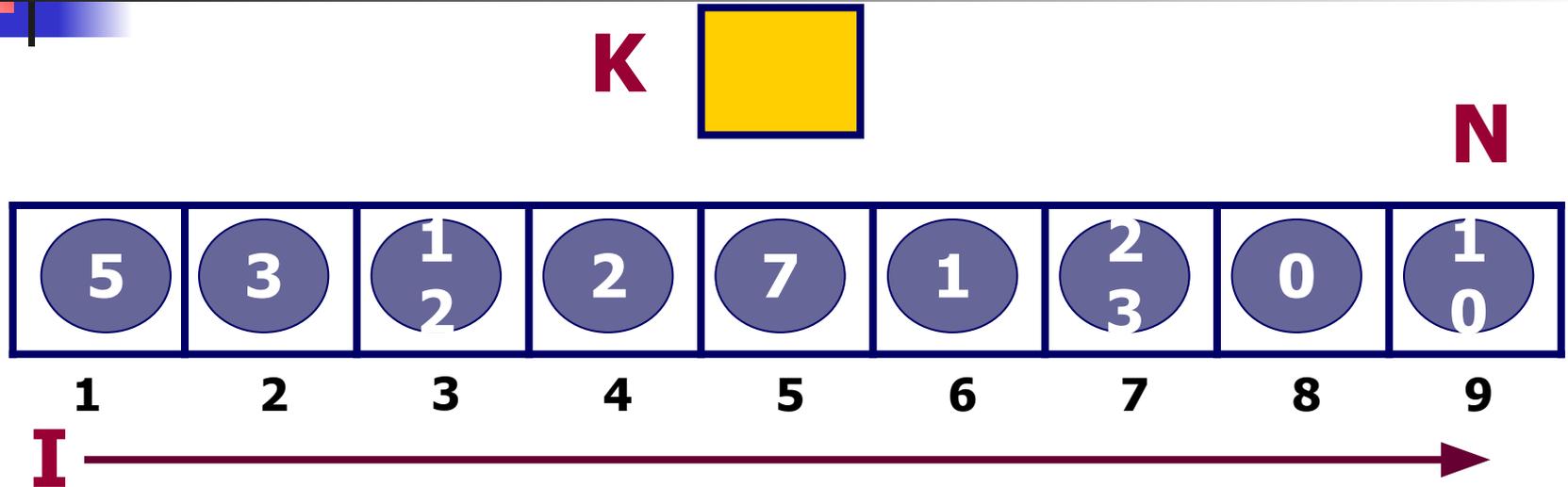
## СОРТИРОВКА ВЫБОРОМ

**Кондраткова**

**Татьяна Алексеевна**

**ГБОУ Лицей № 82 СПб**

# Поиск минимального элемента в массиве

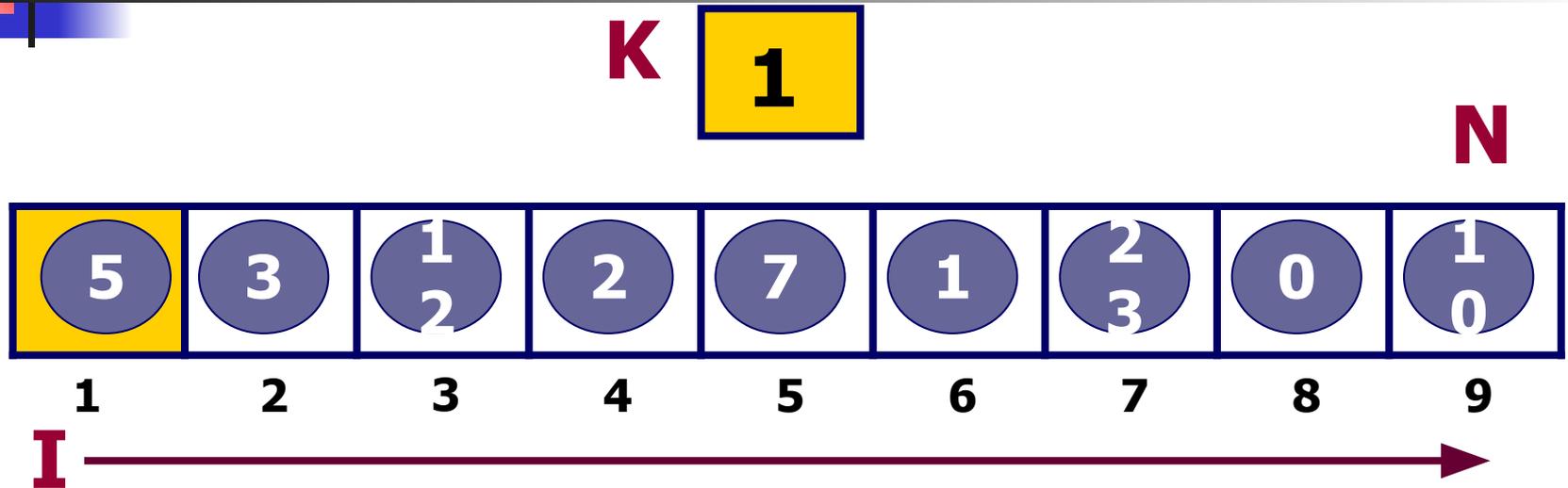


**N** - количество элементов в массиве;

**I** - переменная цикла;

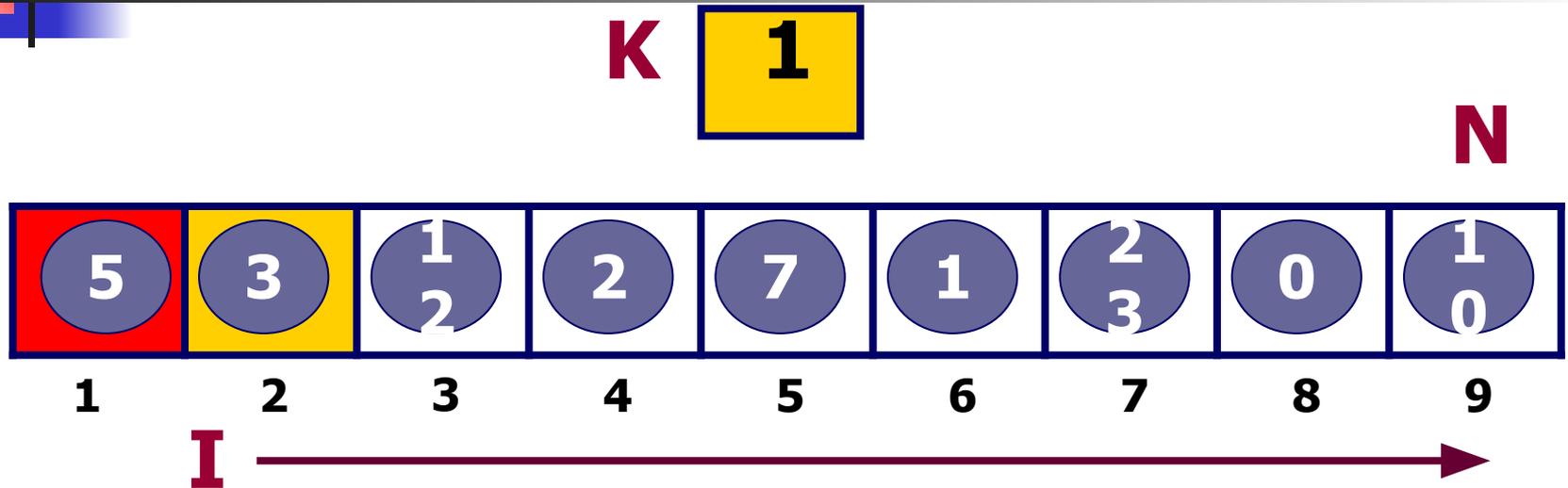
**K** - переменная, в которую записывается индекс (номер) минимального элемента в массиве.

# Поиск минимального элемента в массиве



Индекс первого элемента записывается в переменную **K**.

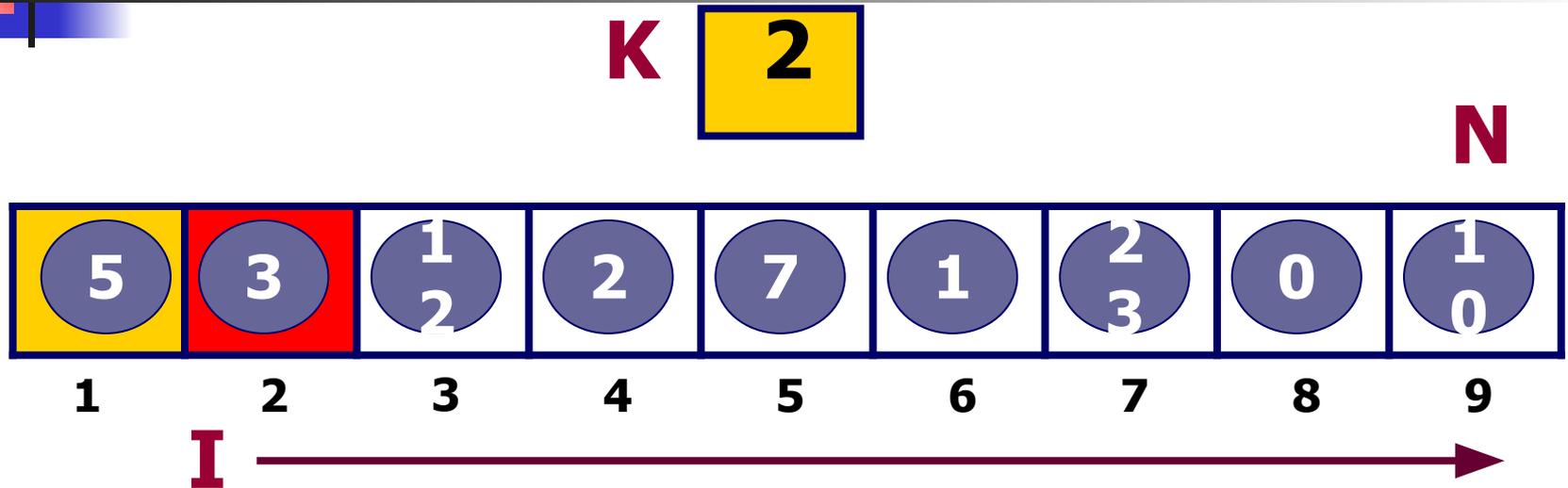
# Поиск минимального элемента в массиве



Сравниваются элемент с индексом **I** (текущий элемент массива) и элемент с индексом **K**.

Индекс меньшего по значению элемента записывается в переменную **K**.

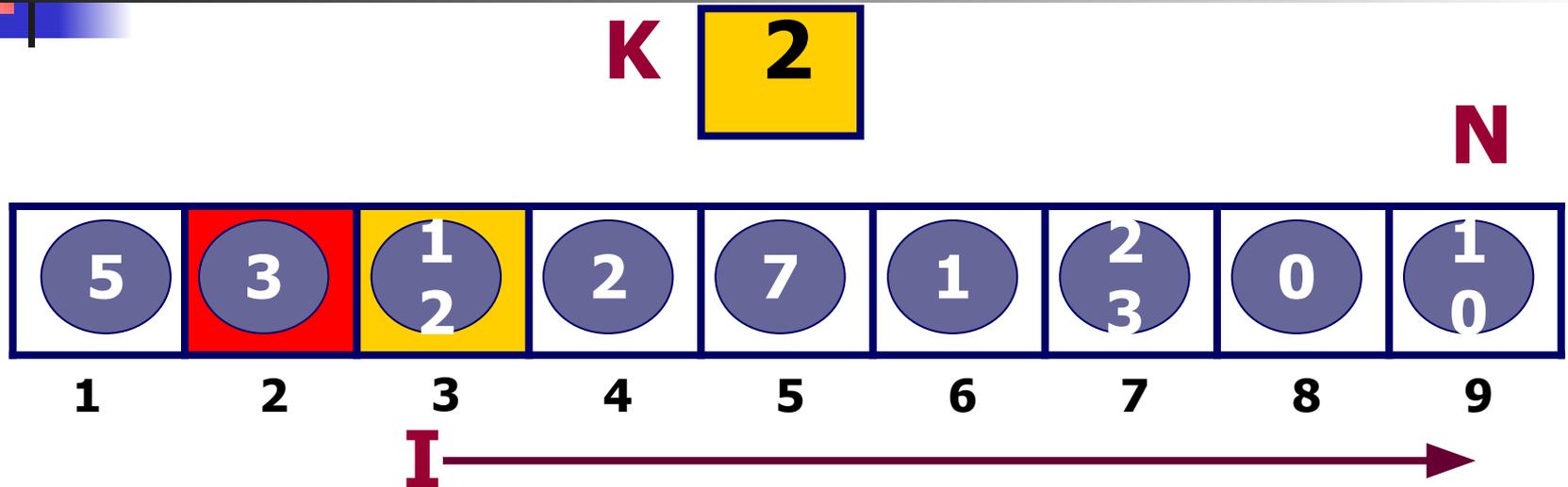
# Поиск минимального элемента в массиве



Сравниваются элемент с индексом **I** (текущий элемент массива) и элемент с индексом **K**.

Индекс меньшего по значению элемента записывается в переменную **K**.

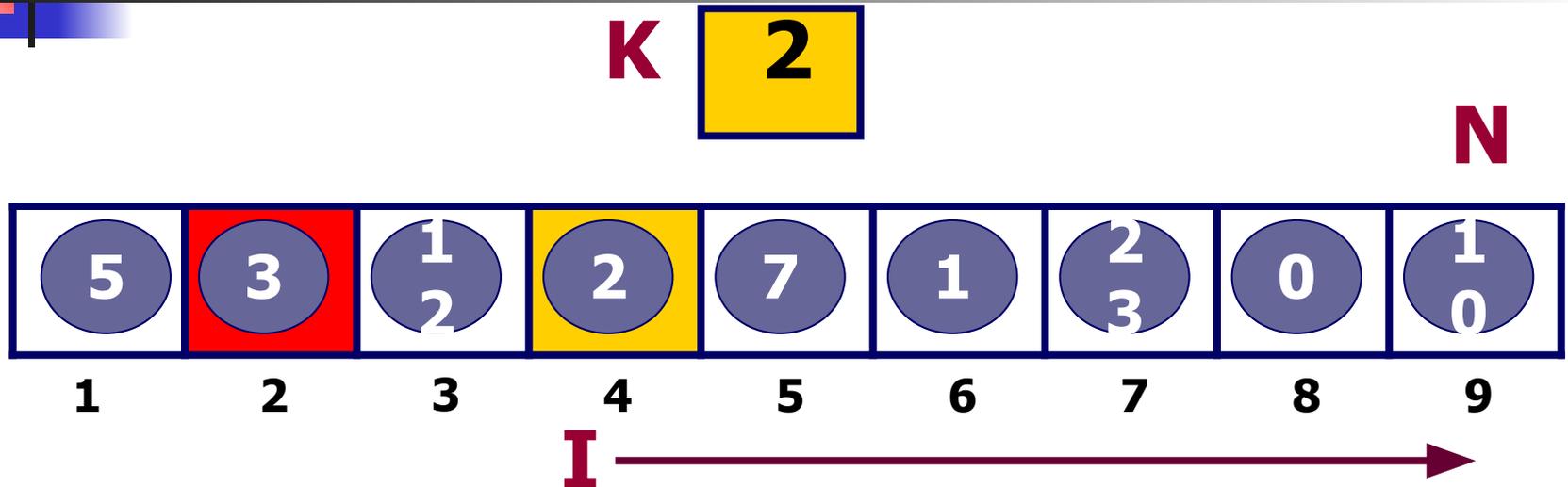
# Поиск минимального элемента в массиве



Сравниваются элемент с индексом **I** (текущий элемент массива) и элемент с индексом **K**.

Индекс меньшего по значению элемента записывается в переменную **K**.

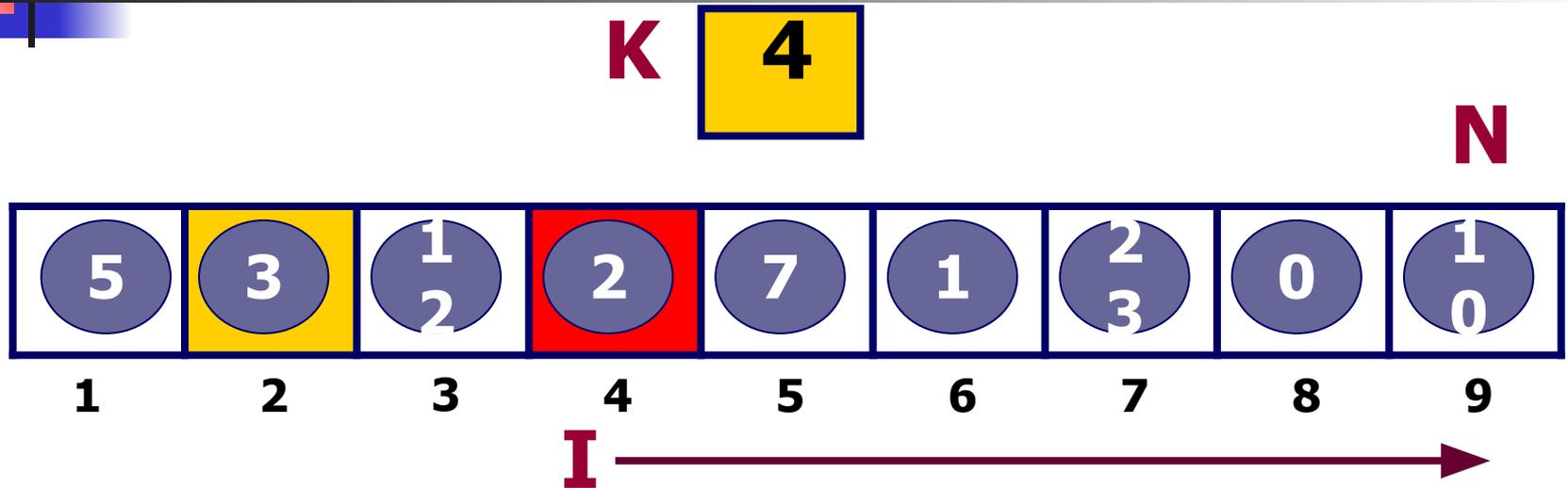
# Поиск минимального элемента в массиве



Сравниваются элемент с индексом **I** (текущий элемент массива) и элемент с индексом **K**.

Индекс меньшего по значению элемента записывается в переменную **K**.

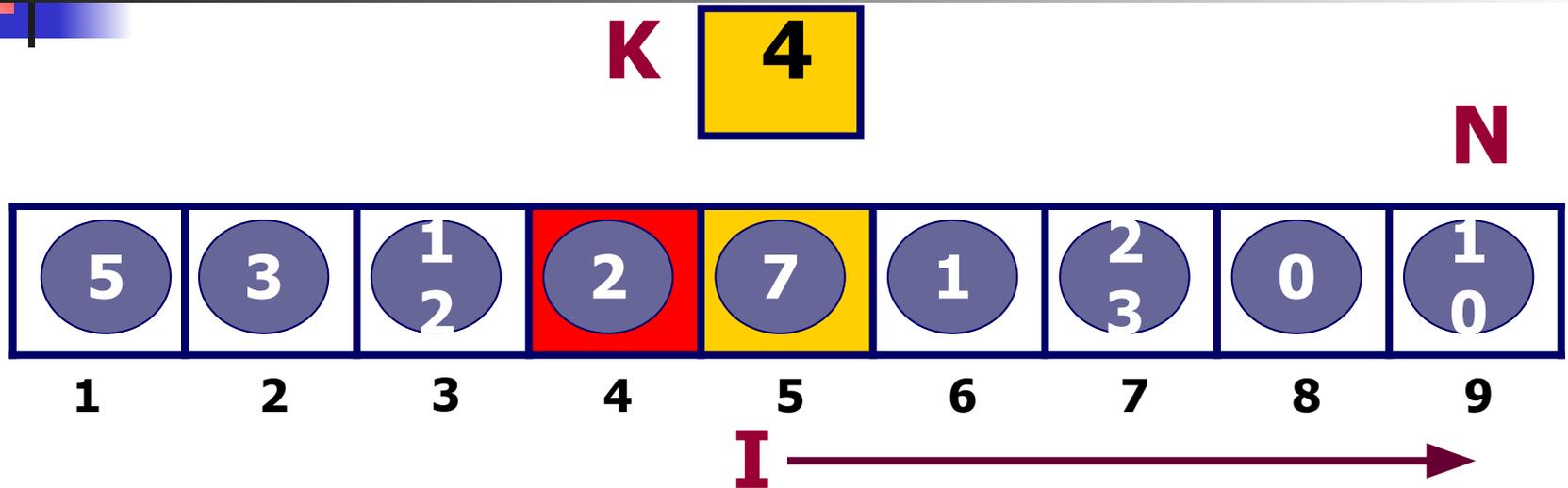
# Поиск минимального элемента в массиве



Сравниваются элемент с индексом **I** (текущий элемент массива) и элемент с индексом **K**.

Индекс меньшего по значению элемента записывается в переменную **K**.

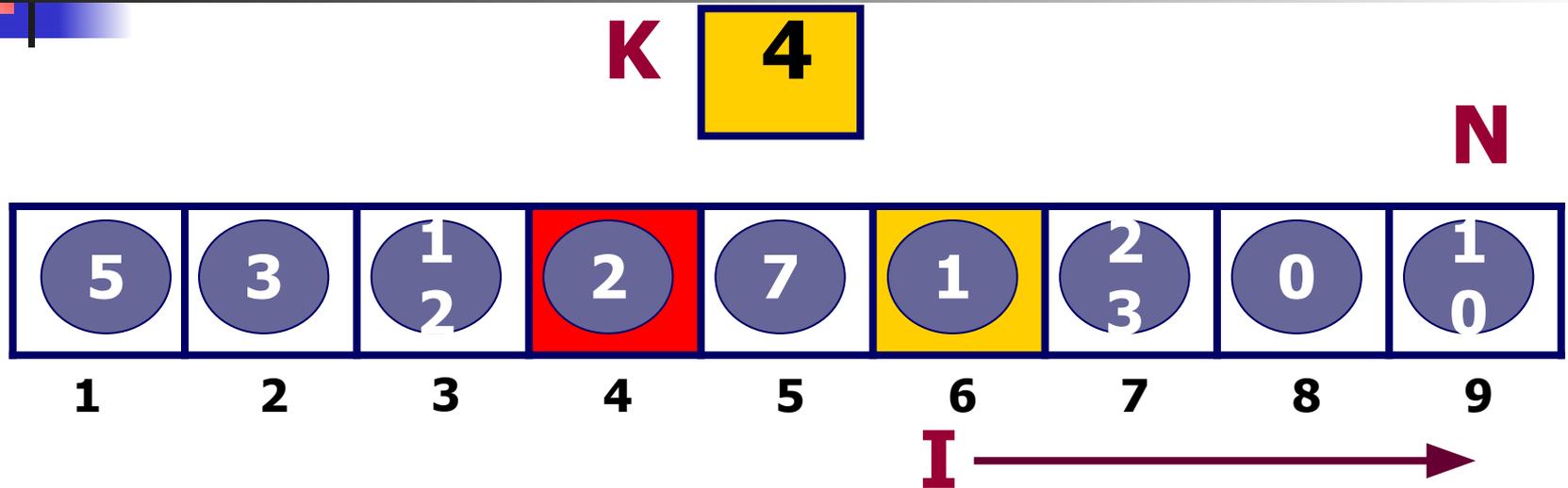
# Поиск минимального элемента в массиве



Сравниваются элемент с индексом **I** (текущий элемент массива) и элемент с индексом **K**.

Индекс меньшего по значению элемента записывается в переменную **K**.

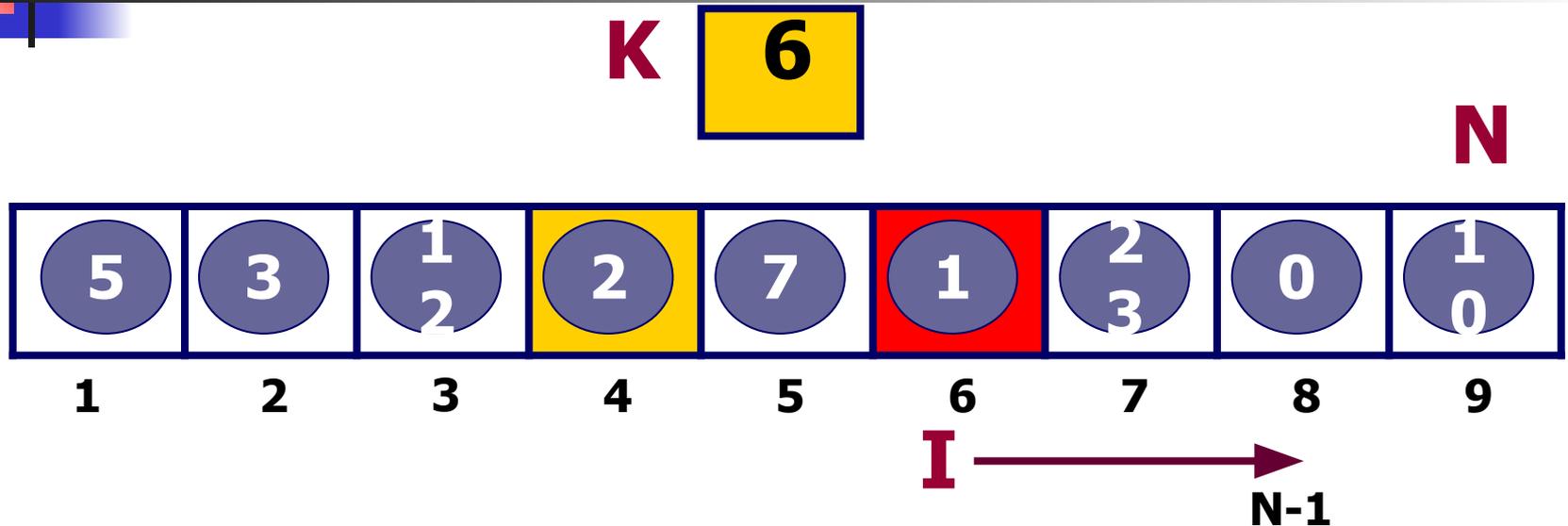
# Поиск минимального элемента в массиве



Сравниваются элемент с индексом **I** (текущий элемент массива) и элемент с индексом **K**.

Индекс меньшего по значению элемента записывается в переменную **K**.

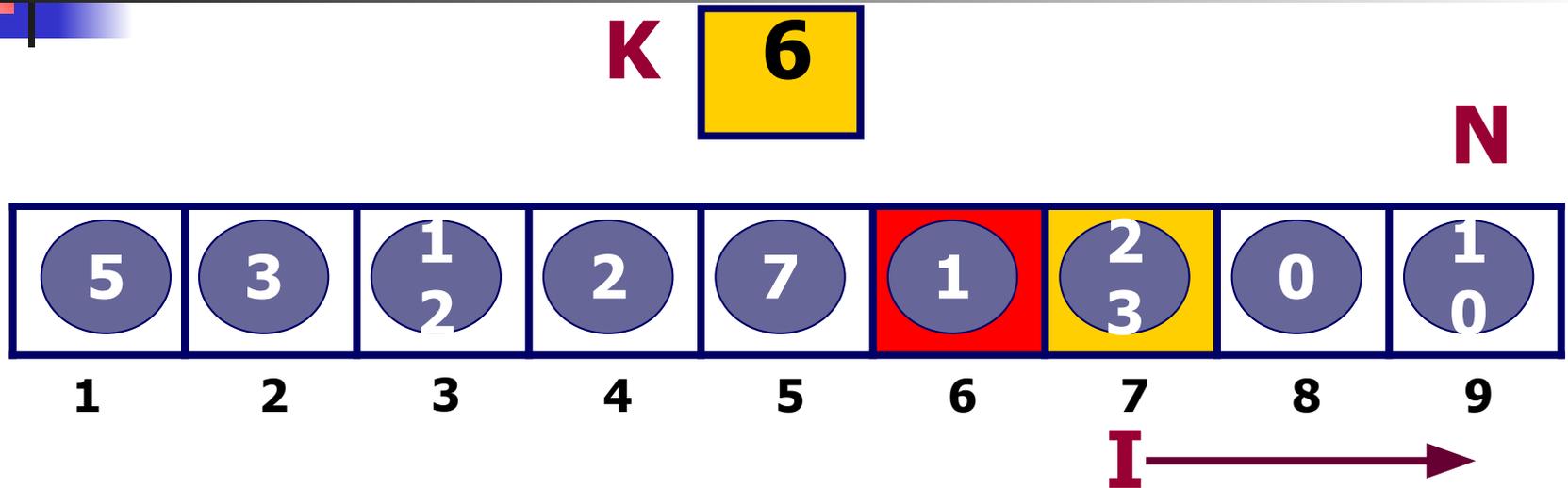
# Поиск минимального элемента в массиве



Сравниваются элемент с индексом **I** (текущий элемент массива) и элемент с индексом **K**.

Индекс меньшего по значению элемента записывается в переменную **K**.

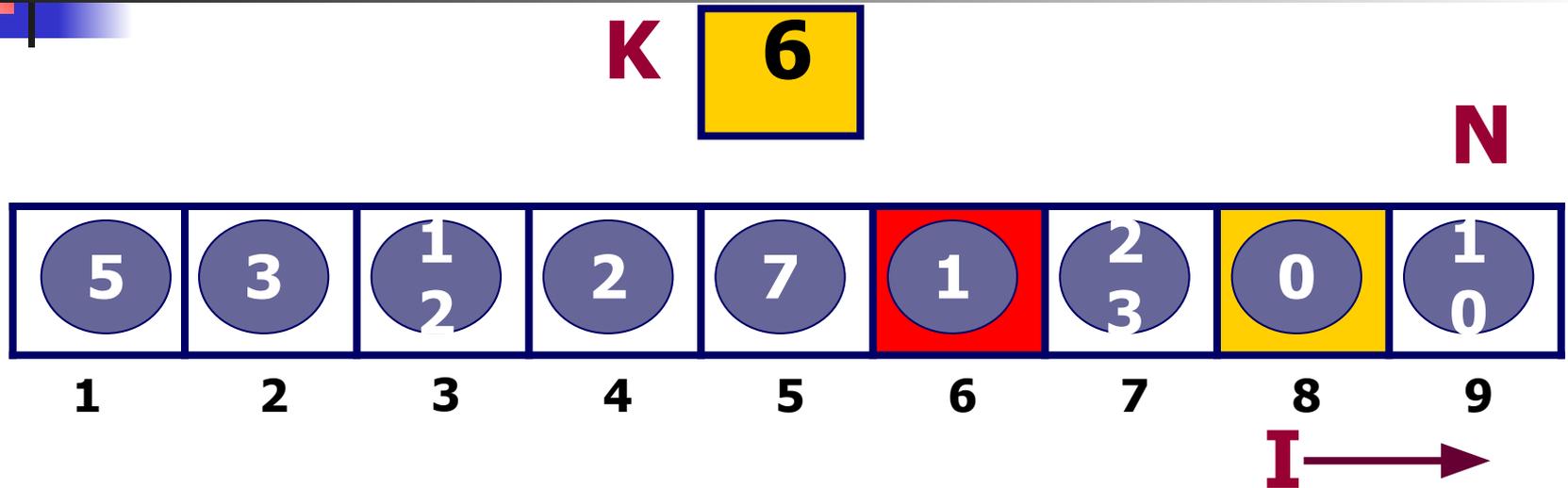
# Поиск минимального элемента в массиве



Сравниваются элемент с индексом **I** (текущий элемент массива) и элемент с индексом **K**.

Индекс меньшего по значению элемента записывается в переменную **K**.

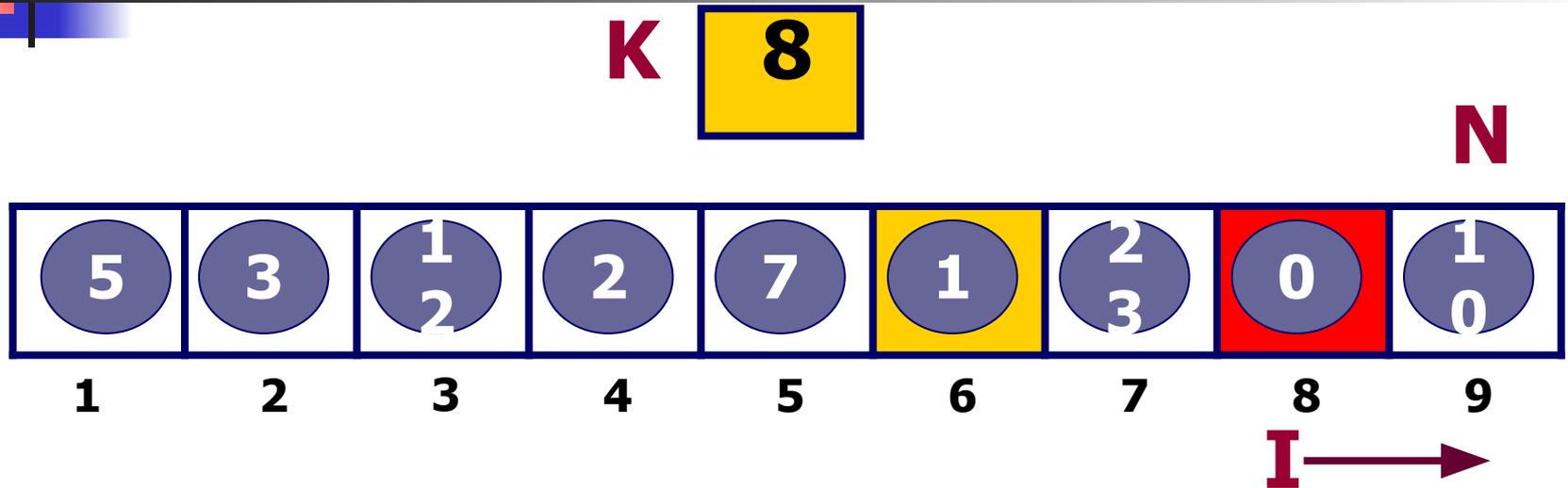
# Поиск минимального элемента в массиве



Сравниваются элемент с индексом **I** (текущий элемент массива) и элемент с индексом **K**.

Индекс меньшего по значению элемента записывается в переменную **K**.

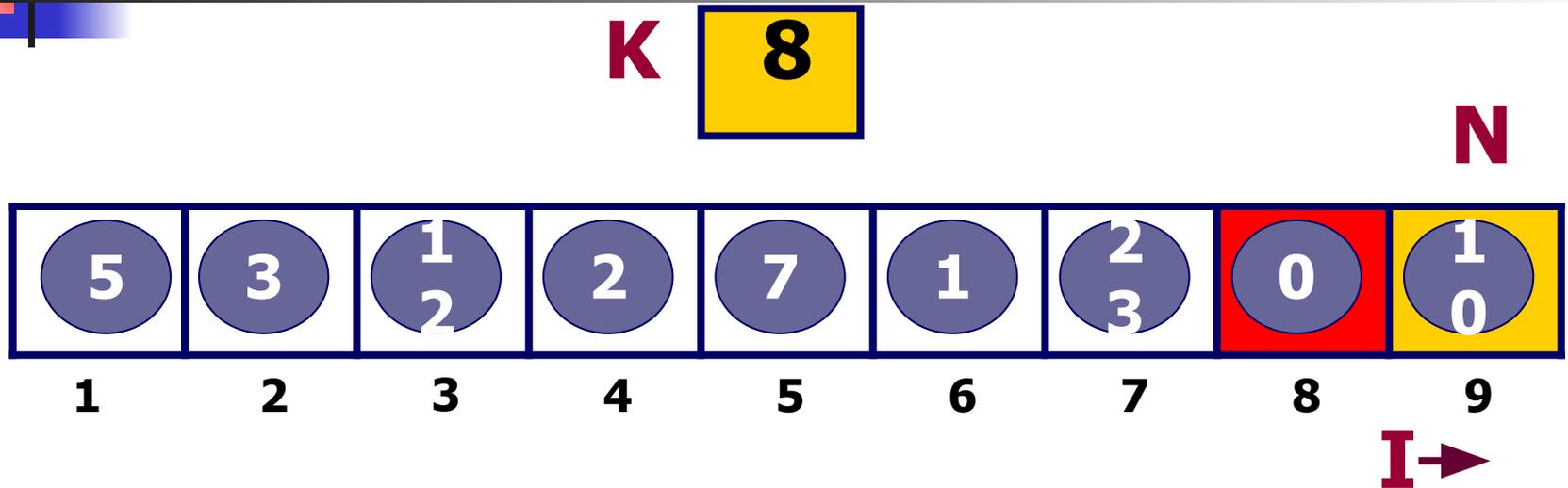
# Поиск минимального элемента в массиве



Сравниваются элемент с индексом **I** (текущий элемент массива) и элемент с индексом **K**.

Индекс меньшего по значению элемента записывается в переменную **K**.

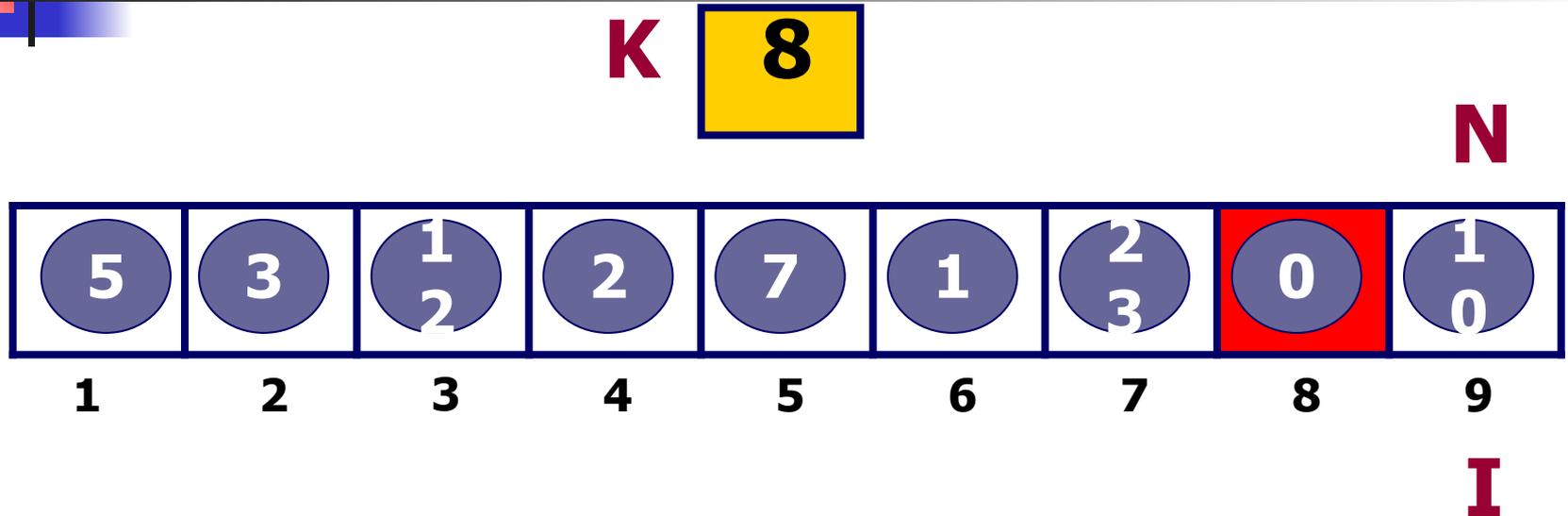
# Поиск минимального элемента в массиве



Сравниваются элемент с индексом **I** (текущий элемент массива) и элемент с индексом **K**.

Индекс меньшего по значению элемента записывается в переменную **K**.

# Поиск минимального элемента в массиве



В переменной **K** записан индекс меньшего по значению элемента массива.

**A[K]** – минимальный элемент массива

**program minmas;** {заголовок программы, не обязателен}

**TYPE** {секция описания типов}

**MASS= array [1..30] of integer;** {объявляется тип}

**var** {секция описания переменных}

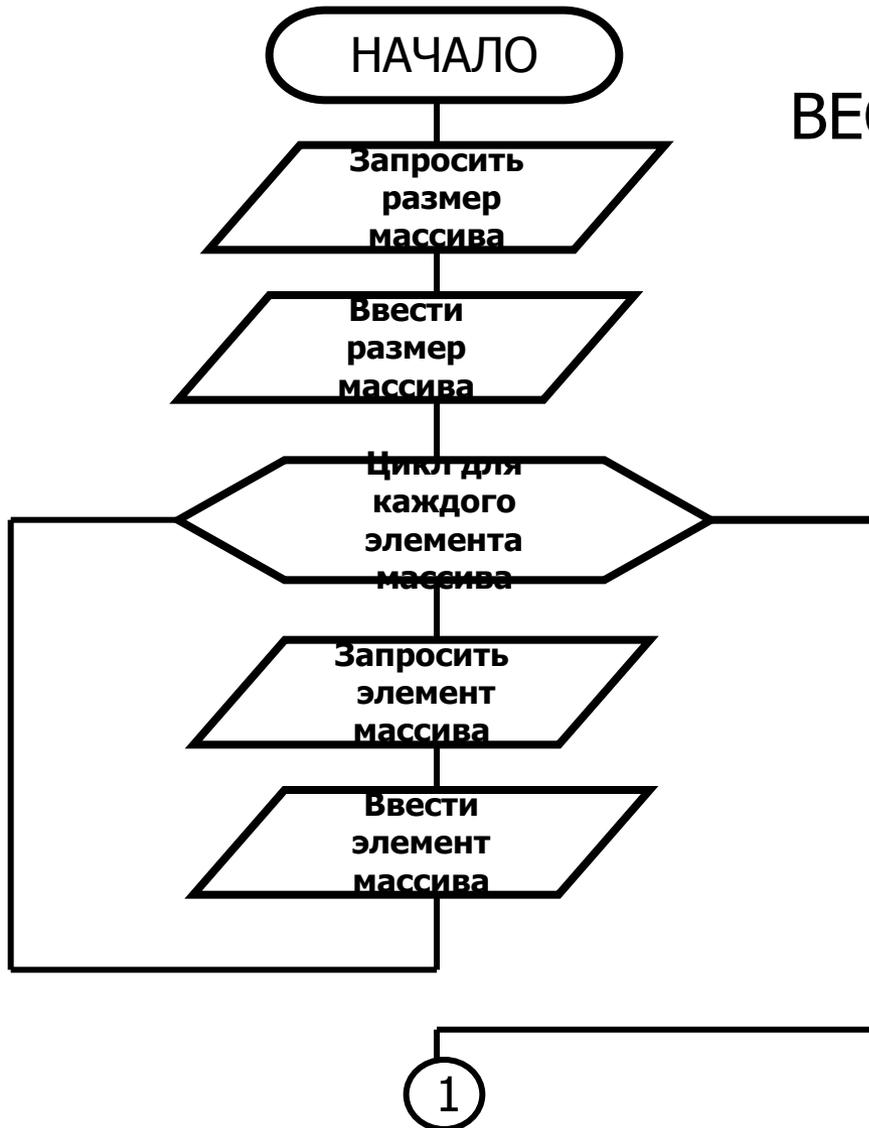
**N:1..30;** {размер массива }

**A: MASS;** {массив из N **целых** чисел}

**I:1..30;** {переменная цикла }

**K:1..30;** {индекс минимального элемента}

# Блок формирования массива



BEGIN

```
Write(' N= ');
```

```
ReadLn(N);
```

```
FOR I:=1 TO N DO
```

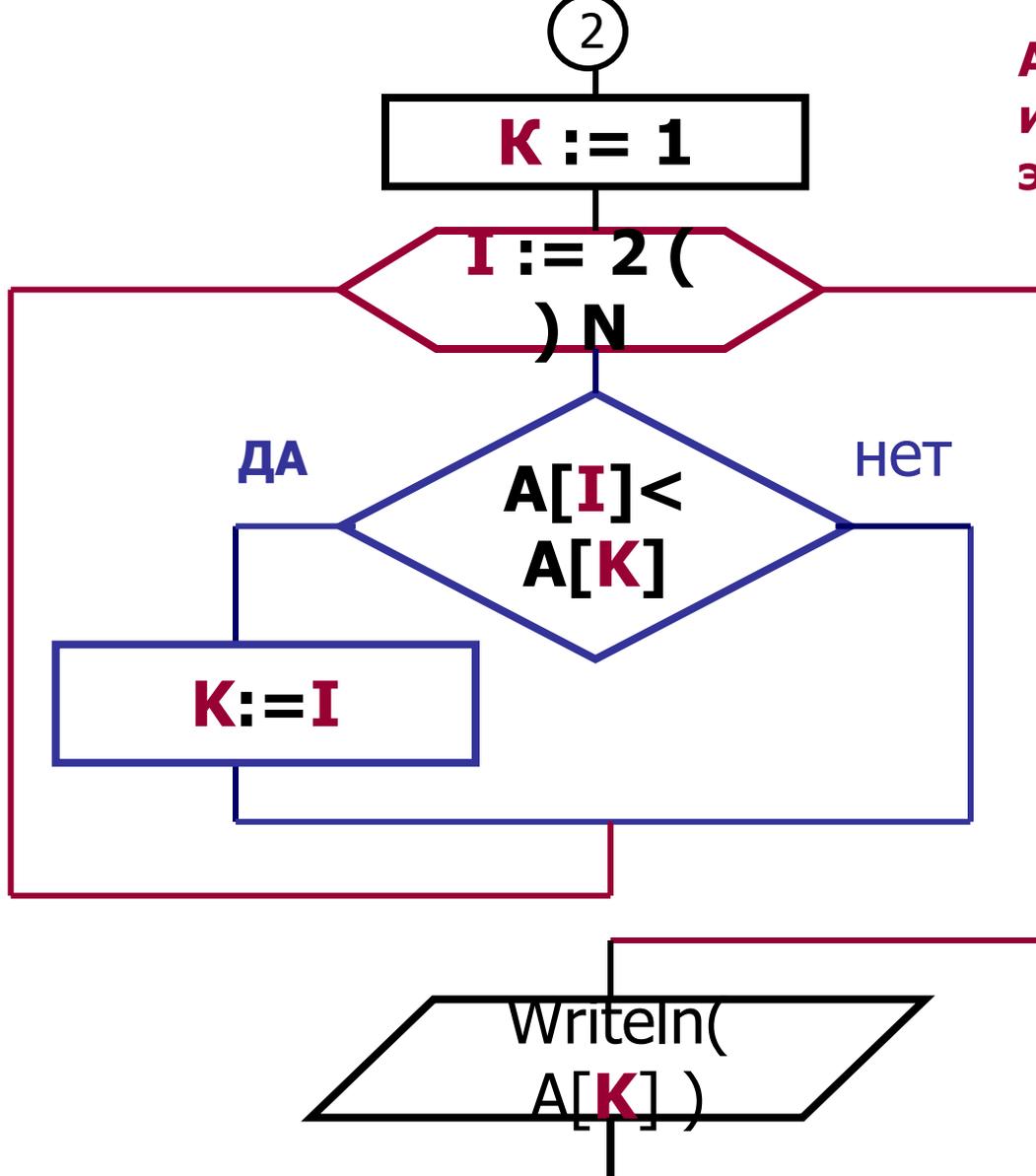
```
begin
```

```
Write(' A[ ', I, ' ]= ');
```

```
ReadLn(A[ I ])
```

```
end;
```

**АЛГОРИТМ ПОИСКА  
индекса минимального  
элемента в массиве**





---

# СОРТИРОВКА МАССИВА МЕТОДОМ ВЫБОРА

# Порядок работы:

## Разработка, отладка и тестирование программы:

### Программа должна:

- Сформировать массив (ввод данных с клавиатуры);
- Вывести массив на экран для просмотра данных;
- Произвести сортировку массива по алгоритму «Метод выбора»;
- Вывести массив на экран для просмотра результата.

### После того, как Вы убедились, что программа работает правильно

### Определить эффективность метода:

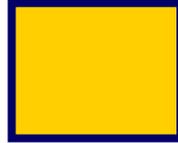
- Поставить счётчики в программу;
- Запустить программу на выполнение;
- Снять показания счётчиков на первом входном массиве;
- Записать показания счётчиков в бланк лабораторной работы;
- Запустить программу и снять показания счётчиков на втором и третьем входных массивах.
- Описать дополнительное рабочее поле ОЗУ в бланке лабораторной работы.

# **РАЗРАБОТКА АЛГОРИТМА**

## **СОРТИРОВКИ МЕТОДОМ ВЫБОРА**

(массив целых чисел сортируется по  
не убыванию элементов)

**K**



**N**



**1**

**2**

**3**

**4**

**5**

**6**

**7**

**8**

**9**

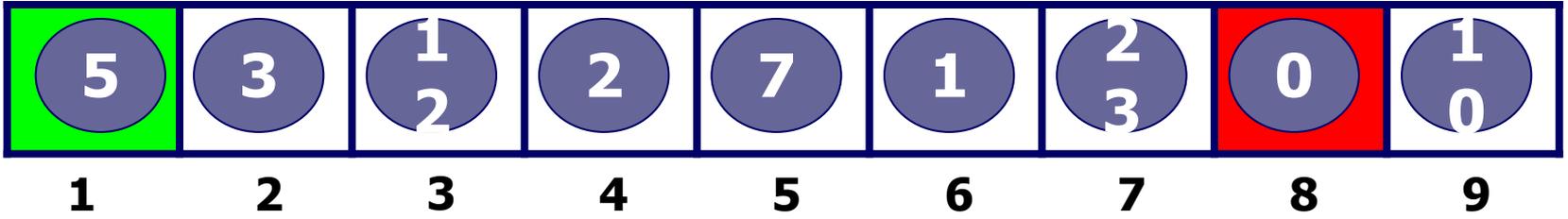
**I**



**J**

**K** 8

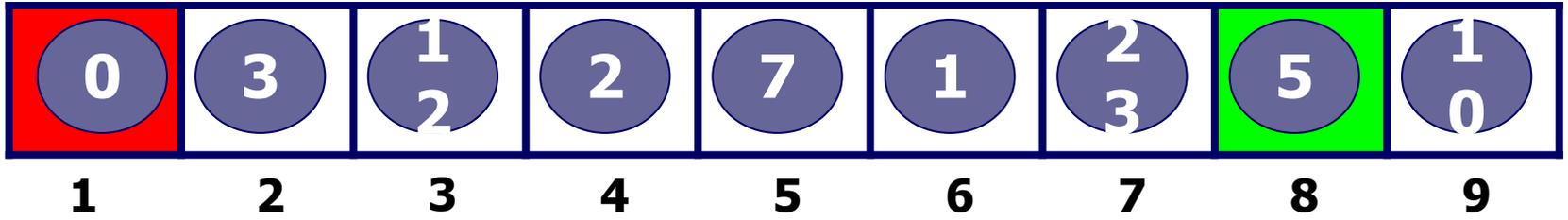
**N**



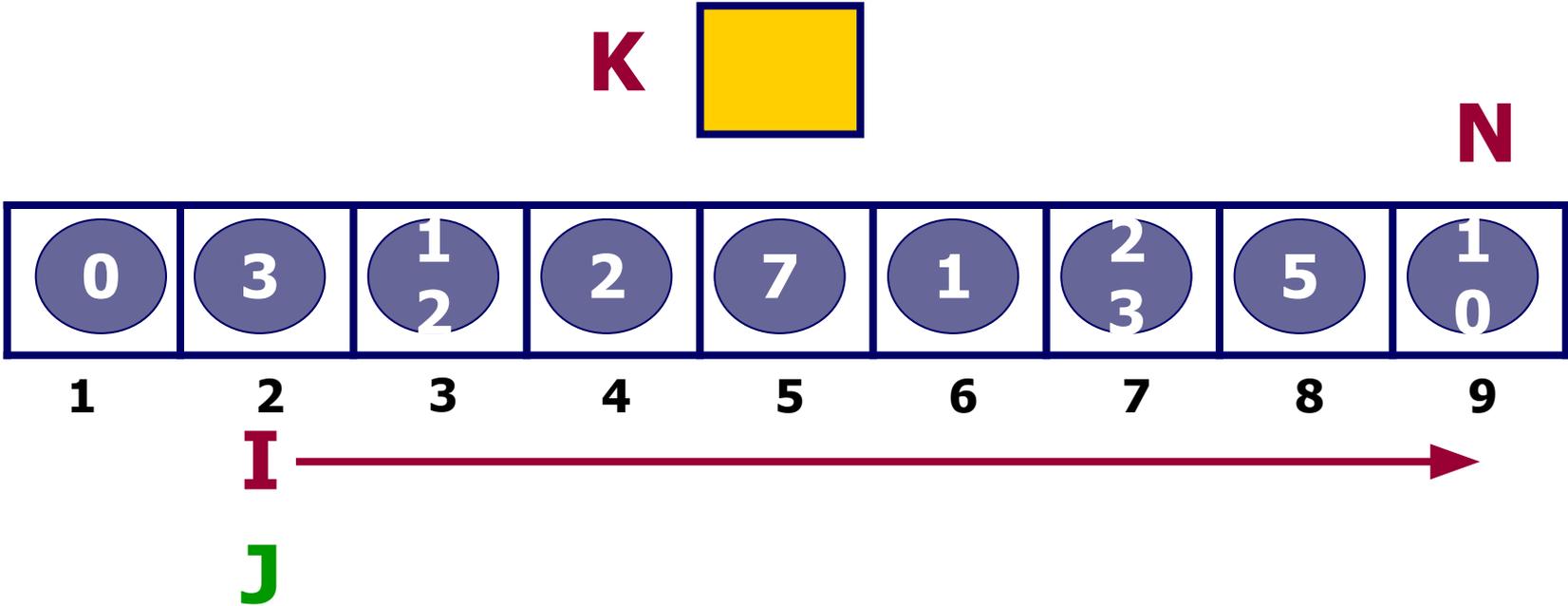
**J**

**K** 8

**N**

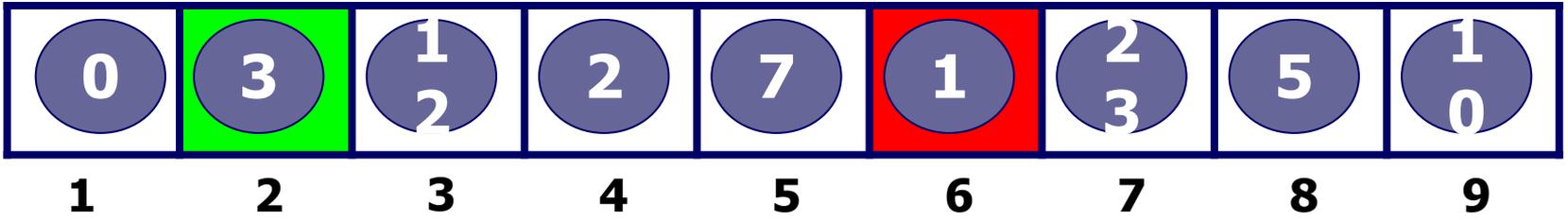


**J**



**K** **6**

**N**



**I**

**J**

**K**

**6**

**N**



**1**

**2**

**3**

**4**

**5**

**6**

**7**

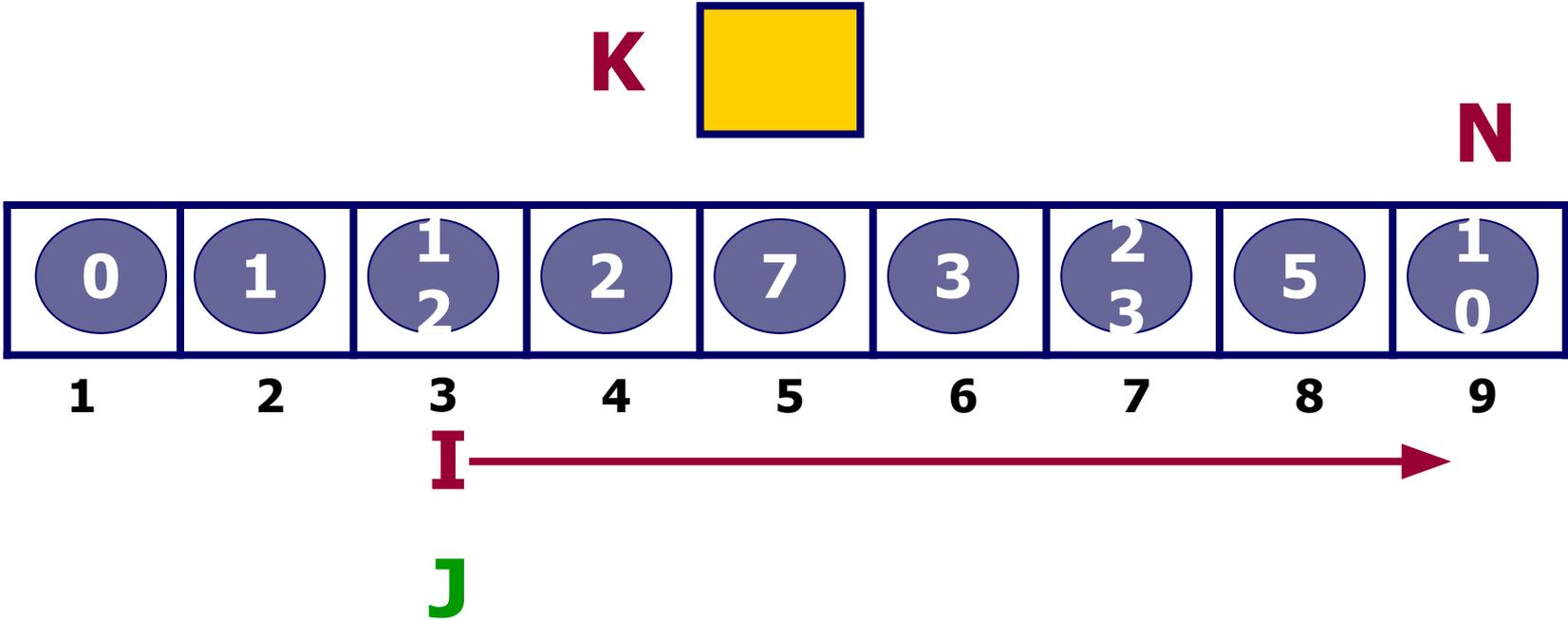
**8**

**9**



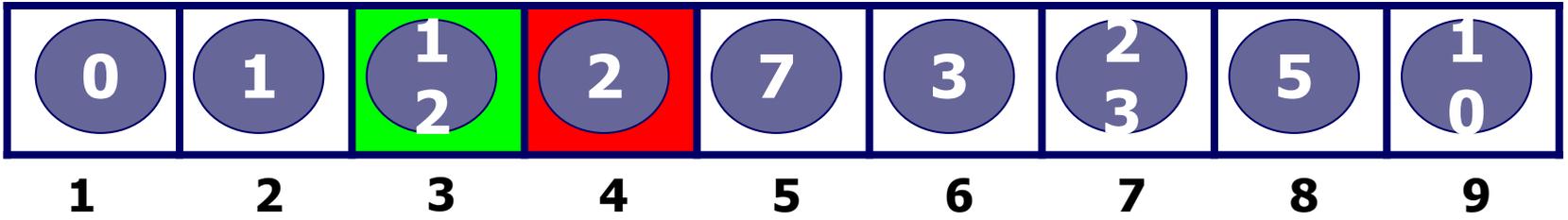
**I**

**J**



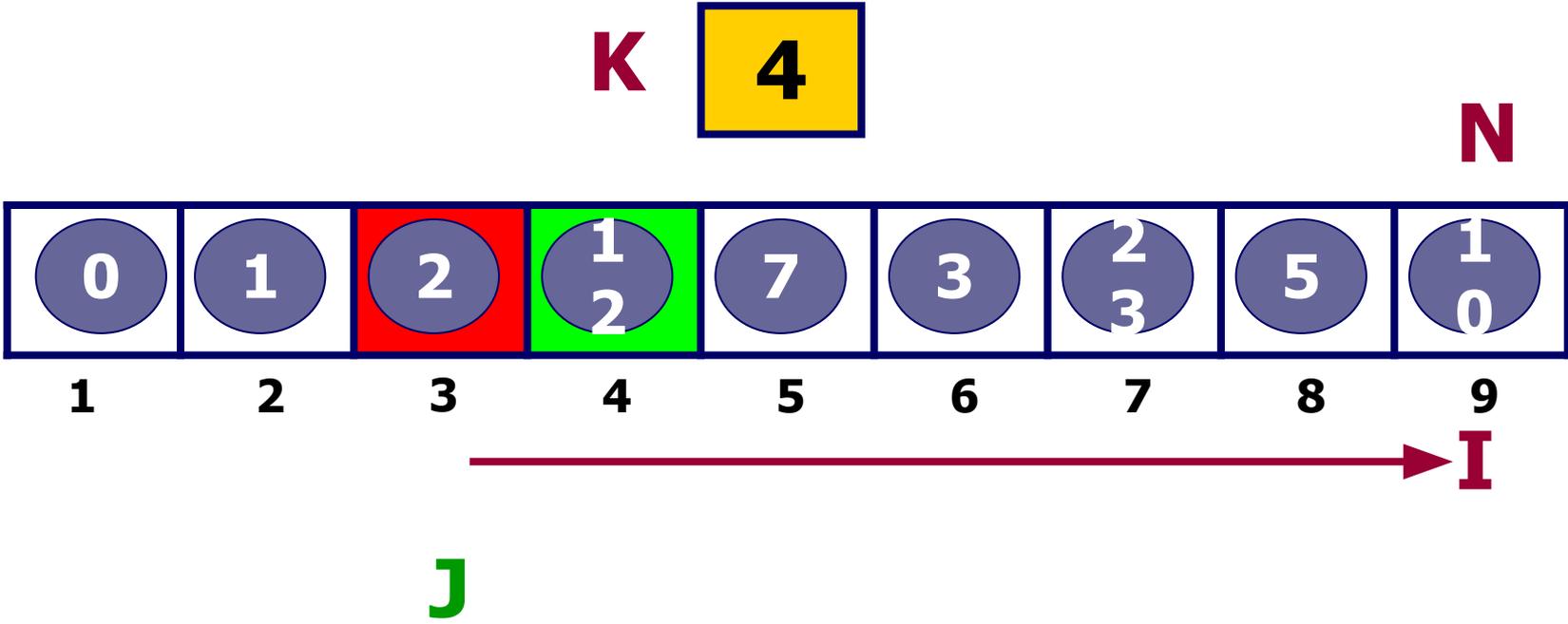
**K** **4**

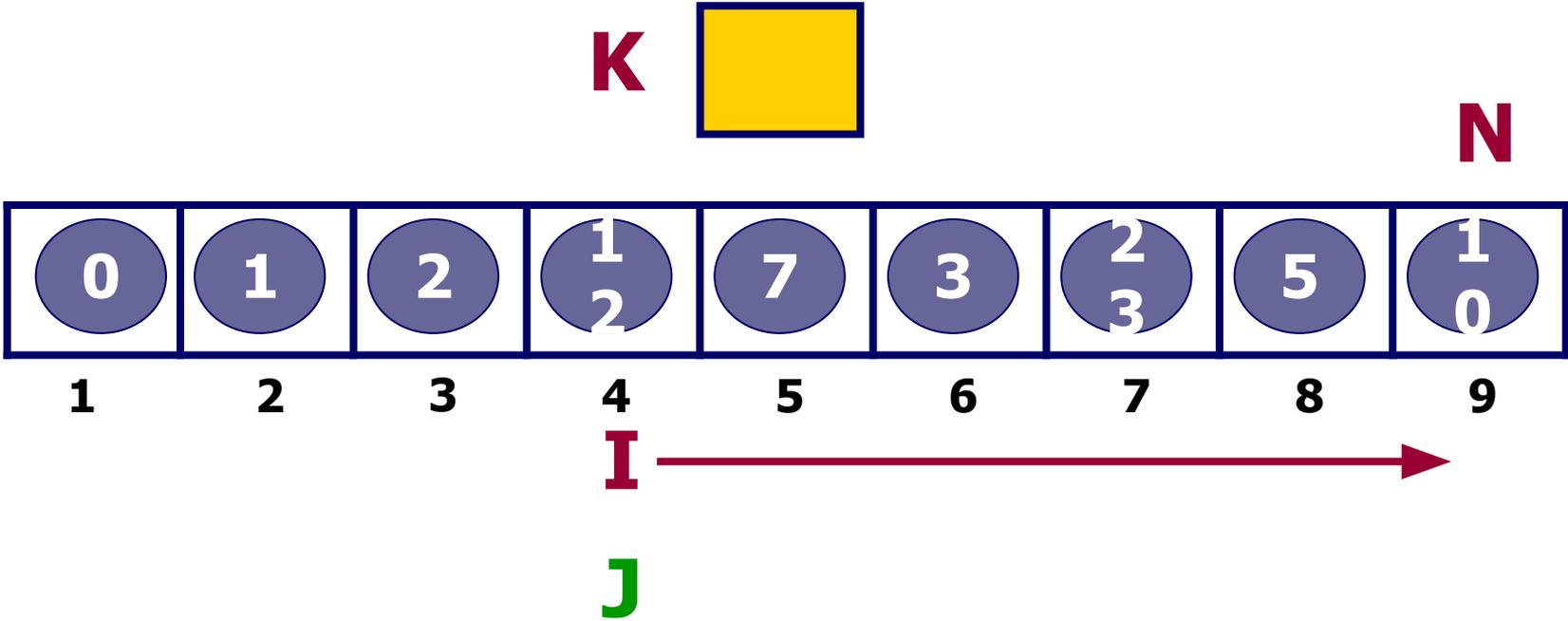
**N**



**I**

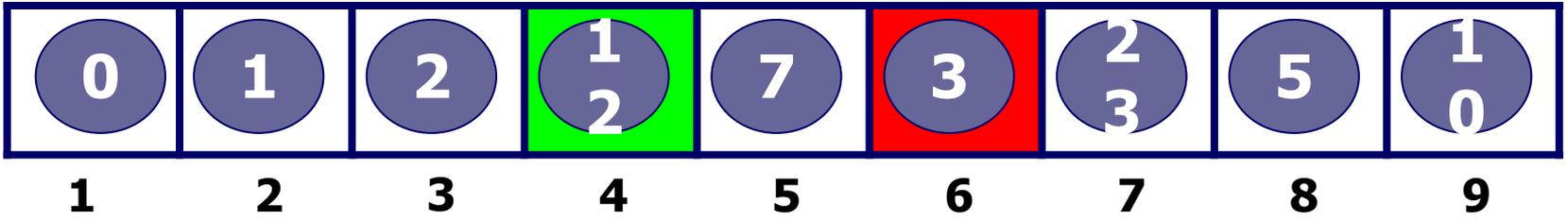
**J**





**K** **6**

**N**

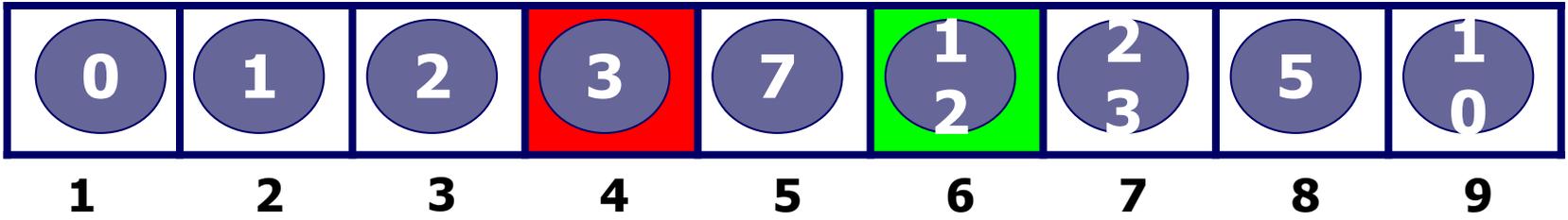


**J**

**K**

**6**

**N**



**I**

**J**

**K**



**N**



**1**

**2**

**3**

**4**

**5**

**6**

**7**

**8**

**9**

**I**



**J**

**K**

**8**

**N**



**1**

**2**

**3**

**4**

**5**

**6**

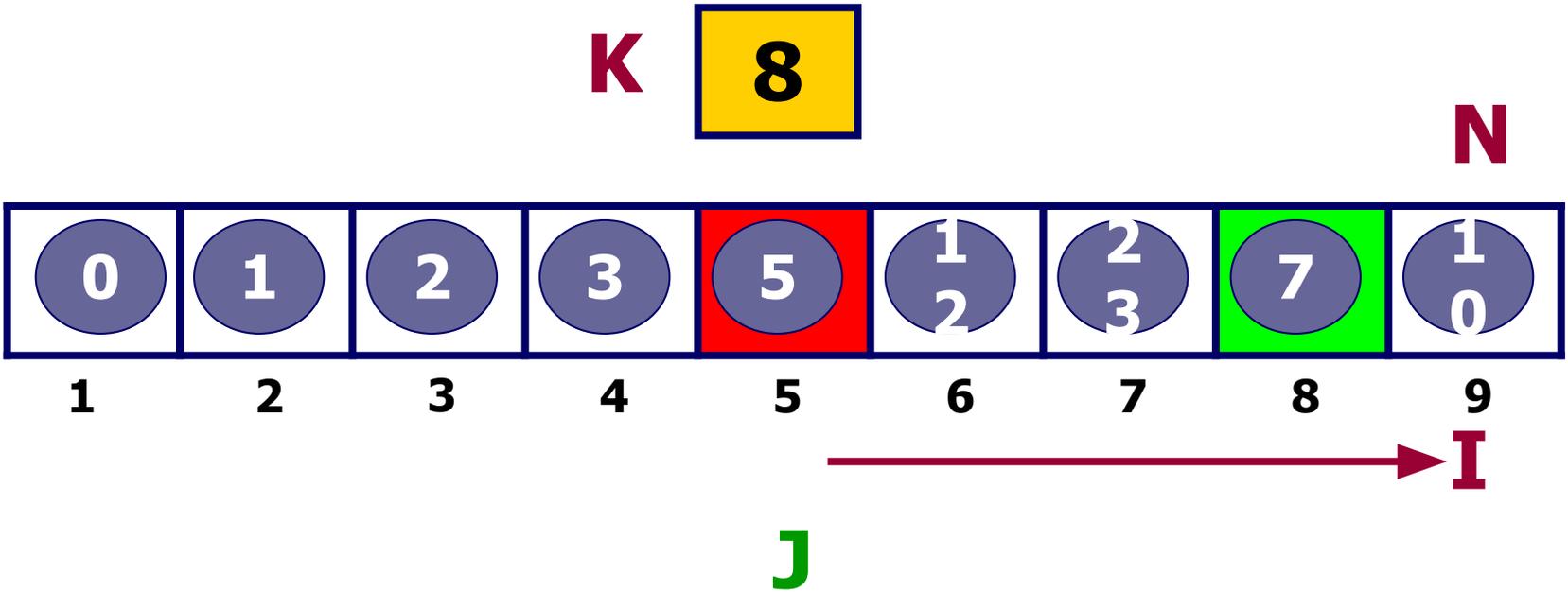
**7**

**8**

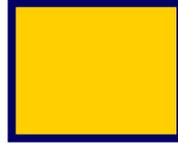
**9**

**I**

**J**



**K**



**N**



**1**

**2**

**3**

**4**

**5**

**6**

**7**

**8**

**9**

**I**



**J**

**K**

**8**

**N**



**1**

**2**

**3**

**4**

**5**

**6**

**7**

**8**

**9**



**I**

**J**

**K**

**8**

**N**



**1**

**2**

**3**

**4**

**5**

**6**

**7**

**8**

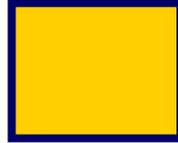
**9**



**I**

**J**

**K**



**N**



**1**

**2**

**3**

**4**

**5**

**6**

**7**

**8**

**9**

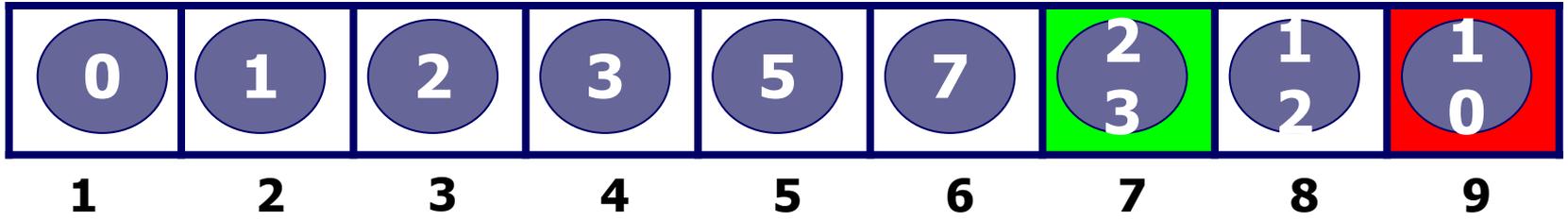
**I**



**J**

**K** **9**

**N**

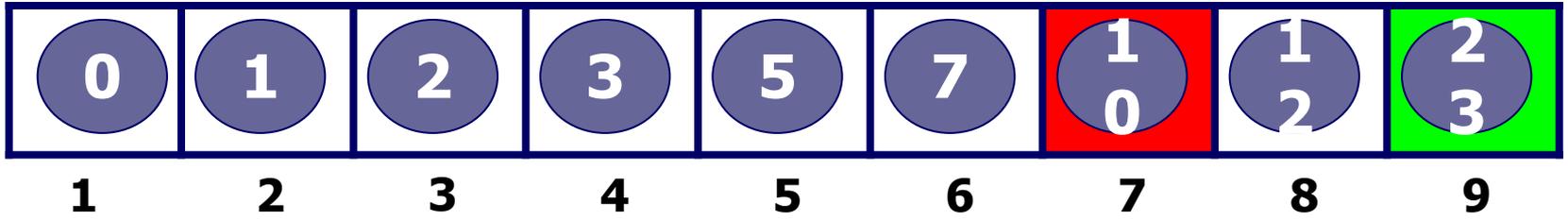


**I**

**J**

**K** **9**

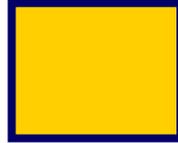
**N**



**I**

**J**

**K**



**N**



**1**

**2**

**3**

**4**

**5**

**6**

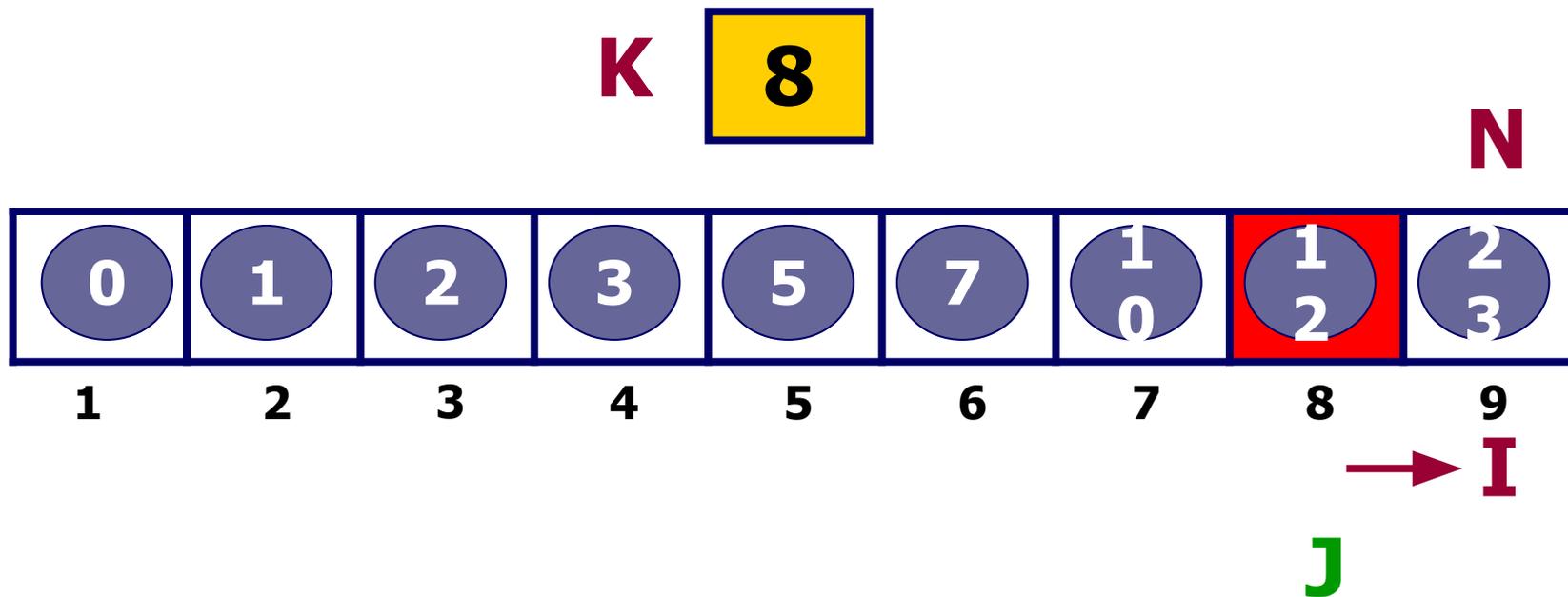
**7**

**8**

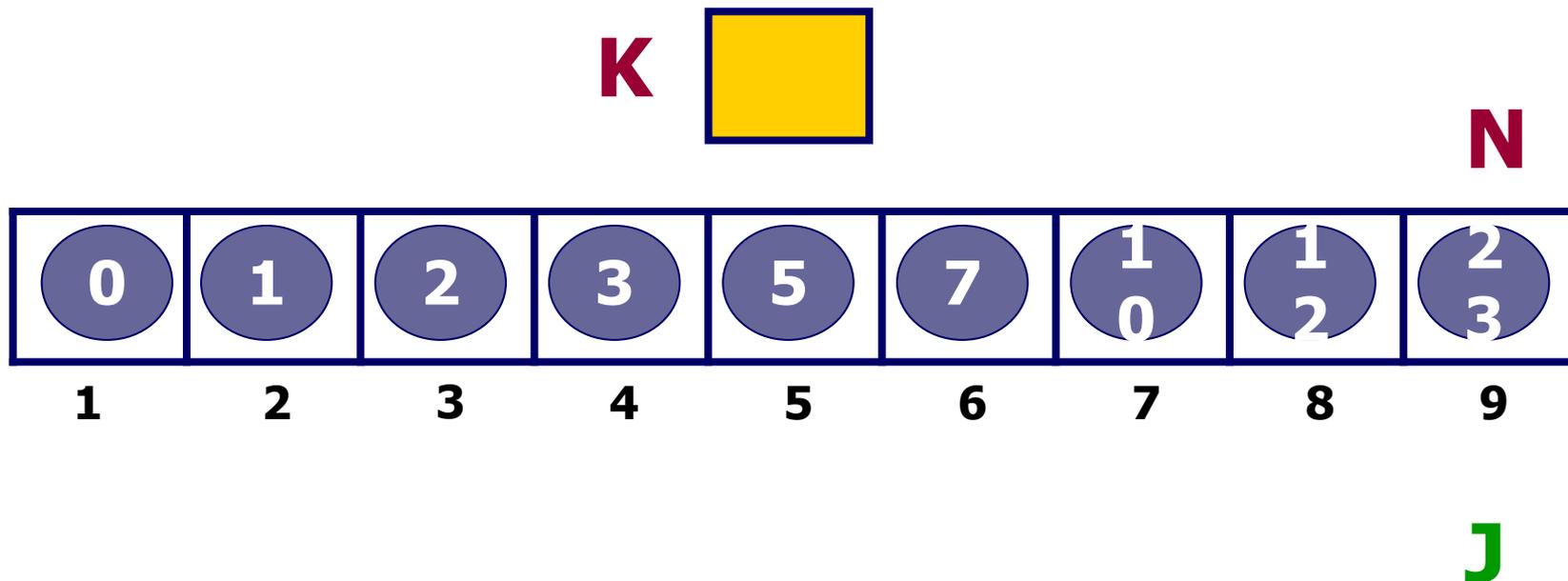
**9**



**J**



Если **K=J**, то обмен не нужно делать.

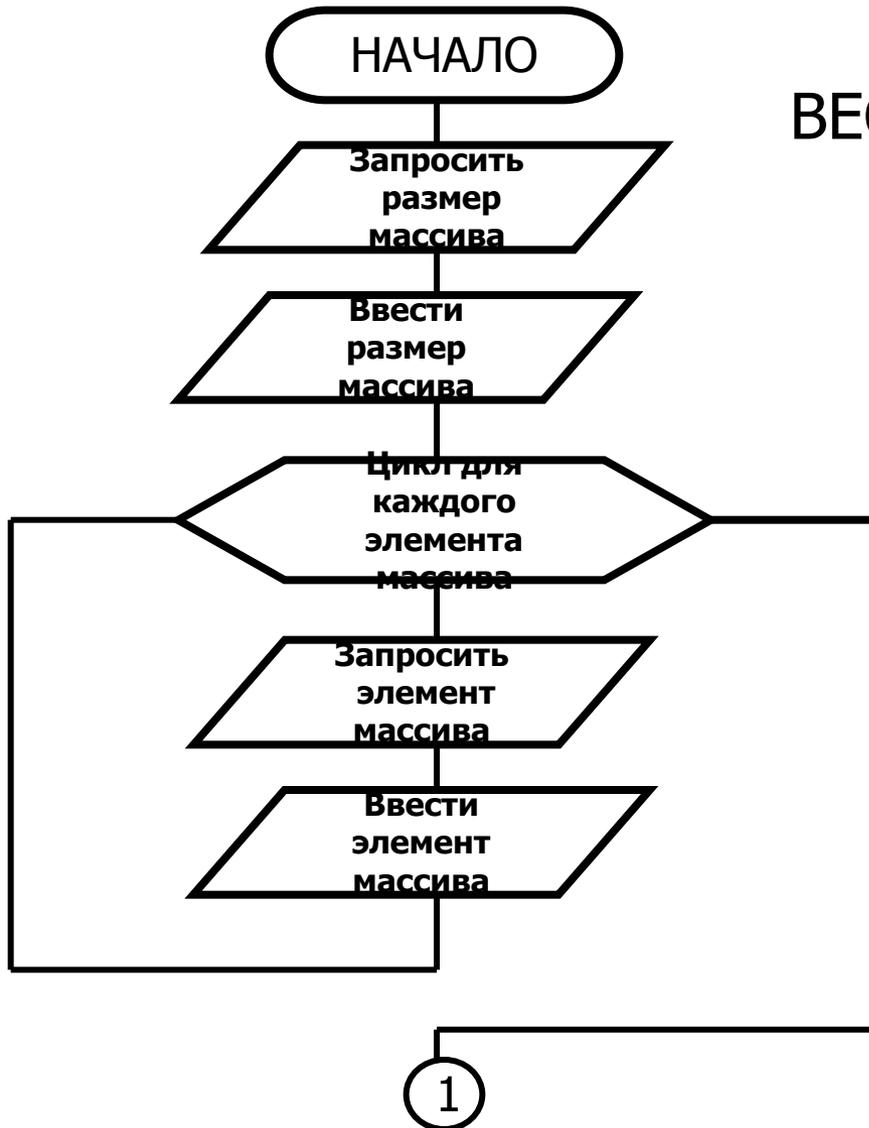


Процесс сортировки завершен за  $N-1$  цикл по переменной **J**.

# Описание переменных

```
program viborsort;    {заголовок программы, не обязателен}
TYPE                {секция описания типов}
  MASS= array [1..30] of integer; {объявляется тип}
var                 {секция описания переменных}
  N:1..30;          {размер массива }
  A: MASS;          {массив из N целых чисел}
  I:1..30;          {переменная цикла для поиска мин. }
  J:1..30;          {переменная внешнего цикла}
  L:integer;        {переменная для обмена}
  K:1..30;          {индекс минимального элемента}
  CS: integer;      {счётчик числа сравнений}
  CP: integer;      {счётчик числа перестановок}
```

# Блок формирования массива



BEGIN

```
Write(' N= ');
```

```
ReadLn(N);
```

```
FOR I:=1 TO N DO
```

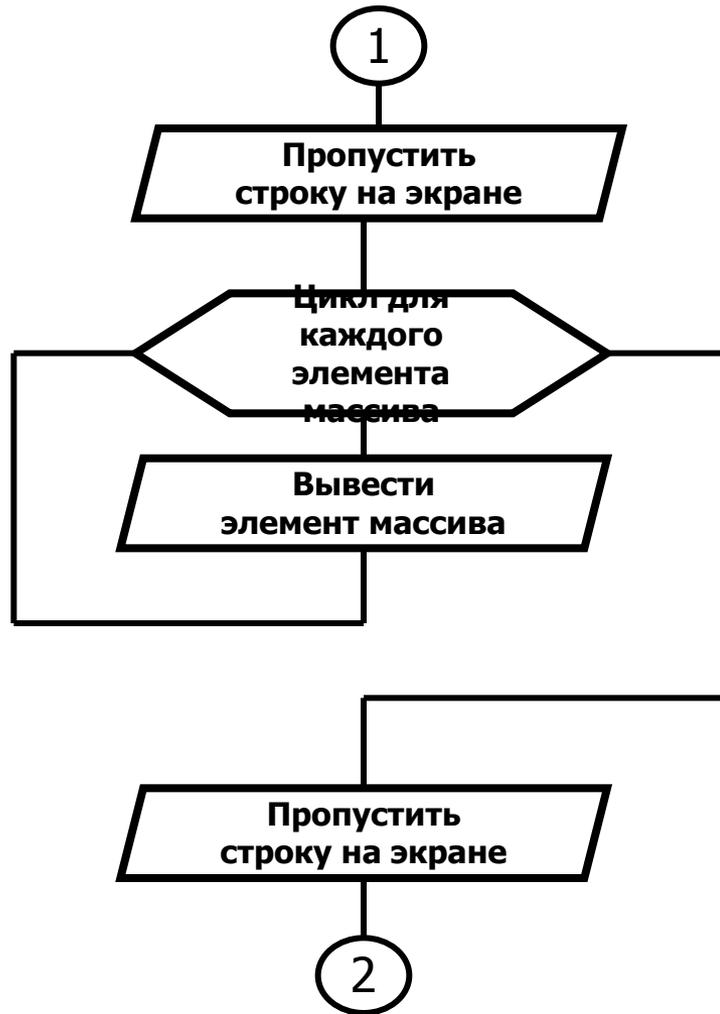
```
begin
```

```
Write(' A[ ', I, ' ]= ');
```

```
ReadLn(A[ I ])
```

```
end;
```

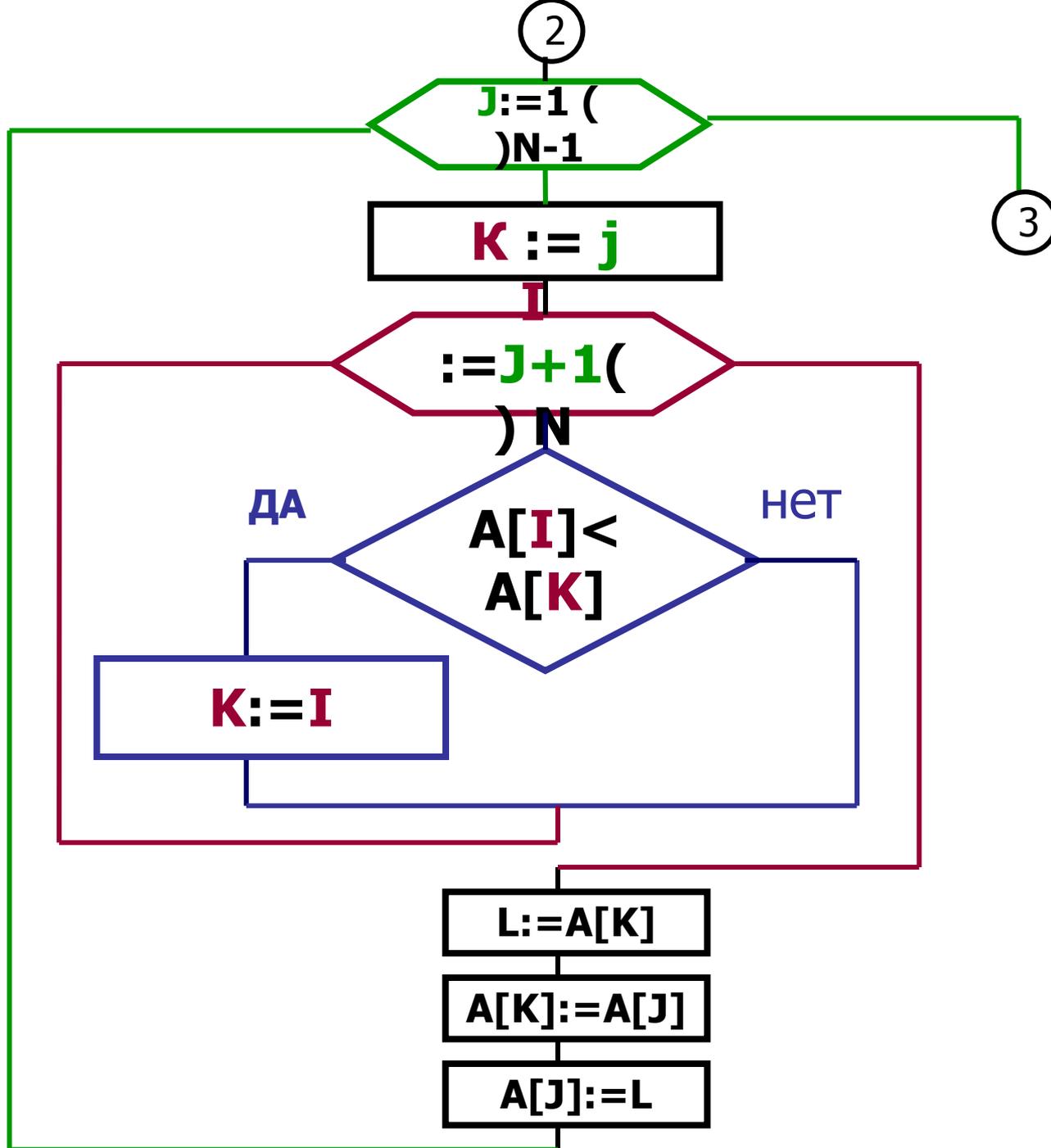
# Блок печати массива



WriteLn;

```
FOR I:=1 TO N DO  
  Write(A[ I ], ' ');
```

WriteLn;



## ОСНОВНАЯ ЧАСТЬ ПРОГРАММЫ

**FOR J:=1 TO N-1 DO**

**BEGIN**

**K:=J;**

**FOR I:=J+1 TO N DO**

**IF A[I]<A[K] THEN K:=I;**

**IF k<> J THEN**

**begin**

**L:=A[K];**

**A[K]:=A[J];**

**A[J]:=L;**

**end;**

**END;**

## Куда ставить счётчики?

```
CS:=0;
CP:=0;
FOR J:=1 TO N-1 DO
  BEGIN
    K:=J;
    FOR I:=J+1 TO N DO
      begin
        CS:=CS+1;
        IF A[I]<A[K] THEN K:=I;
      end;
    IF k<>J THEN
      begin
        L:=A[K];
        A[K]:=A[J];
        A[J]:=L;
        CP:=CP+3;
      end;
  END;
WriteLn(' CS=' ,CS);
WriteLn(' CP=' ,CP);
```

Обнулить счётчики  
до начала сортировки

Увеличить на 1  
значение счётчика  
числа сравнений

Увеличить на 3  
значение счётчика  
числа перестановок

Обратите внимание на то, что  
после добавления оператора в  
тело цикла с параметром  
необходимо поставить  
**операторные скобки.**

Вывести на экран  
значения счётчиков  
после завершения сортировки

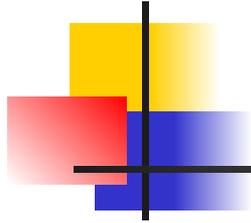
После завершения сортировки ещё раз вывести на экран значения элементов массива, чтобы проверить, что сортировка прошла успешно.



```
WriteLn;
```

```
FOR I:=1 TO N DO  
    Write(A[ I ], ' ');  
ReadLn;
```

```
END. { конец программы }
```



## **Внимание!**

**Переменные-счётчики нужны только для проведения эксперимента.**

**Они не влияют на алгоритм сортировки и во время сортировки не задействованы.**

**Эти переменные не должны учитываться как дополнительная рабочая память.**