

## Задача 2. Разбиение массива

Источник:	базовая*
Имя входного файла:	input.txt
Имя выходного файла:	output.txt
Ограничение по времени:	1 секунда
Ограничение по памяти:	разумное

В первых четырёх байтах входного файла задано целое число  $N$  — количество чисел в массиве. В следующих четырёх байтах записано целое число  $p$  — пивот-элемент. Далее идут  $N$  четырёхбайтовых целых чисел — содержимое массива. Все числа знаковые, длина последовательности лежит в диапазоне:  $1 \leq N \leq 10^6$ .

Требуется реализовать функцию разбиения массива относительно заданного пивот-элемента с сигнатурой:

```
//partitions array a[0..n-1] into two subarrays, returning value k
// the subarray a[0..k-1] must have all elements <= pivot
// the subarray a[k..n-1] must have all elements >= pivot
int partition(int *a, int n, int pivot);
```

и применить её к заданной в файле последовательности. Внутри функции разрешается использовать дополнительную память.

**Важно:** Заметим, что элементы, которые в точности равны `pivot`, можно помещать как в левую, так и в правую часть массива. В данной задаче требуется распределить эти элементы примерно поровну. Если в левую часть попадает  $u$  элементов, равных пивоту, а в правую часть —  $v$  элементов, то должно выполняться:  $|u - v| \leq 1$ .

В первые 4 байта выходного файла нужно вывести целое число  $k$  — сколько элементов попадает в левую часть массива. Далее нужно вывести  $N$  четырёхбайтовых целых чисел: содержимое массива  $a$  после выполнения функции `partition`.

### Пример

input.txt															
09	00	00	00	04	00	00	00	06	00	00	00	F8	FF	FF	FF
09	00	00	00	F8	FF	FF	FF	FA	FF	FF	FF	05	00	00	00
02	00	00	00	09	00	00	00	FF	FF	FF	FF				
output.txt															
05	00	00	00	F8	FF	FF	FF	F8	FF	FF	FF	FA	FF	FF	FF
02	00	00	00	FF	FF	FF	FF	05	00	00	00	06	00	00	00
09	00	00	00	09	00	00	00								