

# Partition

## (partition.cpp/c)

Time Limit : 1 sec , Memory Limit : 131072 KB

Quick sort is based on the Divide-and-conquer approach. In QuickSort(A, p, r), first, a procedure Partition(A, p, r) divides an array A[p..r] into two subarrays A[p..q-1] and A[q+1..r] such that each element of A[p..q-1] is less than or equal to A[q], which is, in turn, less than or equal to each element of A[q+1..r]. It also computes the index q.

In the conquer processes, the two subarrays A[p..q-1] and A[q+1..r] are sorted by recursive calls of QuickSort(A, p, q-1) and QuickSort(A, q+1, r).

Your task is to read a sequence A and perform the Partition based on the following pseudocode:

```
Partition(A, p, r)
1 x = A[r]
2 i = p-1
3 for j = p to r-1
4     do if A[j] <= x
5         then i = i+1
6             exchange A[i] and A[j]
7 exchange A[i+1] and A[r]
8 return i+1
```

Note that, in this algorithm, Partition always selects an element A[r] as a pivot element around which to partition the array A[p..r].

## Input (partition.in)

The first line of the input includes an integer  $n$ , the number of elements in the sequence A.

In the second line,  $A_i$  ( $i = 1, 2, \dots, n$ ), elements of the sequence are given separated by space characters.

## Output (partition.out)

Print the sorted sequence. Two contiguous elements of the sequence should be separated by a space character. The element which is selected as the pivot of the partition should be indicated by [ ].

## Constraints

- $1 \leq n \leq 100,000$
- $0 \leq A_i \leq 100,000$

## Sample Input 1

---

```
12
13 19 9 5 12 8 7 4 21 2 6 11
```

## Sample Output 1

---

```
9 5 8 7 4 2 6 [11] 21 13 19 12
```