

# Problem F: Sorting K Window Sums

Filename: ksums

Time limit: 6 seconds

Given a sequence of  $n$  integers, we define a  $k$ -window to be  $k$  contiguous terms in the sequence. There are  $n - k + 1$   $k$ -windows in a sequence of  $n$  integers so long as  $n \geq k$ . We can label each of these windows with their starting index, with the leftmost window having the label 1. Naturally, we can define the sum of a specific  $k$ -window to be the sum of the terms within the window. We can sort each of the  $n - k + 1$   $k$ -windows in a sequence of  $n$  integers based upon the sum of each window, with windows with larger sums coming first. If two  $k$ -windows have the same sum, we break ties by having the one with the lower label come first.

## Problem

Given a sequence of  $n$  integers and a value of  $k$ , determine the sorted order of each of the  $k$ -windows of the sequence.

## Input

The first line will consist of a single positive integer,  $c$ , representing the number of test cases to process. The next line will contain two space separated integers,  $n$  and  $k$ , representing the length of the input sequence and the size of the windows, respectively. The next line will contain  $n$  space separated integers  $x_i$  ( $1 \leq x_i \leq 10^9$ ), representing the  $i^{\text{th}}$  value in the sequence.

## Output

For each case, output  $n - k + 1$  space separated integers representing the sorted order (as defined above) of all of the  $k$ -windows, where each integer is the label of the window represented.

## Input Bounds and Corresponding Credit

30 Points	70 Points
<ul style="list-style-type: none"><li>• <math>1 \leq c \leq 10</math></li><li>• <math>1 \leq n \leq 1000</math></li><li>• <math>1 \leq k \leq n</math></li><li>• <math>1 \leq x_i \leq 10^9</math></li></ul>	<ul style="list-style-type: none"><li>• <math>1 \leq c \leq 15</math></li><li>• <math>1 \leq n \leq 10^5</math></li><li>• <math>1 \leq k \leq n</math></li><li>• <math>1 \leq x_i \leq 10^9</math></li></ul>

## Samples

Input	Output
2 5 3 2 3 4 3 2 10 4 14 12 18 19 13 12 14 14 10 20	2 1 3 1 2 3 4 7 5 6

**Sample Explanation:** In the first input sample, window number 1 contains 2,3 and 4 with a sum of 9, window number 2 contains 3,4 and 3 with a sum of 10, and window number 3 contains 4,3 and 2 with a sum of 9. Since window number 2 has the maximal sum, this window comes first and 2 is the first output. Then, since windows 1 and 3 have the same sum (9), we break the tie by window number, placing window 1 before window 3.