Subarray Sum

Limits: 4s, 512 MB

You are given an array having **N** integers. You have to select at most **K** positions in the array and replace the values at those positions with **0** such that the sum of the subarray which has the maximum sum among all subarrays is maximized. A subarray is a non-empty sequence of consecutive elements of the array.

Input

The first line of the input contains an integer **T** denoting the number of test cases. Then the description of the **T** test cases follows. First line of each of the test cases start with two integers **N** and **K**. It is followed by a line having **N** integers denoting the array.

Constraints:
For all tasks:
1 <= T <= 10
| value of a number in the array | <= 100000

Easy Sub-Task:
0 <= K <= 50
1 <= N <= 50

Medium Sub-Task:
0 <= K <= 1000
1 <= N <= 1000

Hard Sub-Task:
0 <= K <= 5000
1 <= N <= 5000

Output

For every test case output one integer in a separate line in the format **"Case x: y"** where **x** is the number of the test case starting from **1** and **y** is the sum of the subarray which has the maximum sum among all subarrays that you can get after replacing at most **K** numbers with **0**.

Samples

Input	Output
2	Case 1: 15
5 3	Case 2: 3
1 2 3 4 5	
7 2	
1 -4 1 -10 -11 2 -6	

In the first test case we don't need to replace any number with 0.

In the second case we can replace -10 and -11 to get the following array: 1 -4 1 0 0 2 -6. The subarray [1 0 0 2] has the maximum sum which is equal to 3.