

Practice Round APAC test 2017

[A. Lazy Spelling Bee](#)

[B. Robot Rock Band](#)

[C. Not So Random](#)

D. Sums of Sums

[Questions asked](#)

Submissions

Lazy Spelling Bee

5pt	Not attempted 698/1266 users correct (55%)
8pt	Not attempted 496/685 users correct (72%)

Robot Rock Band

6pt	Not attempted 480/622 users correct (77%)
14pt	Not attempted 142/407 users correct (35%)

Not So Random

11pt	Not attempted 204/310 users correct (66%)
20pt	Not attempted 109/158 users correct (69%)

Sums of Sums

8pt	Not attempted 230/395 users correct (58%)
28pt	Not attempted 13/128 users correct (10%)

Top Scores

Jayam	100
Seter	100
KillswitcherEngag...	100
onepunchman	100
Sumeet.Varma	100
gdragon007	100
libenchao	100
jpravishAA	100
vaibhav227	100
wrong	100

Problem D. Sums of Sums

This contest is open for practice. You can try every problem as many times as you like, though we won't keep track of which problems you solve. Read the [Quick-Start Guide](#) to get started.

Small input
8 points

Solve D-small

Large input
28 points

Solve D-large

Problem

Alice presented her friend Bob with an array of N positive integers, indexed from 1 to N . She challenged Bob with many queries of the form "what is the sum of the numbers between these two indexes?" But Bob was able to solve the problem too easily.

Alice took her array and found all $N*(N+1)/2$ non-empty subarrays of it. She found the sum of each subarray, and then sorted those values (in nondecreasing order) to create a new array, indexed from 1 to $N*(N+1)/2$. For example, for an initial array [2, 3, 2], Alice would generate the subarrays [2], [3], [2], [2, 3], [3, 2], and [2, 3, 2] (note that [2, 2], for example, is **NOT** a subarray). Then she'd take the sums -- 2, 3, 2, 5, 5, 7 -- and sort them to get a new array of [2, 2, 3, 5, 5, 7].

Alice has given the initial array to Bob, along with Q queries of the form "what is the sum of the numbers from index L_i to R_i , inclusive, in the new array?" Now Bob's in trouble! Can you help him out?

Input

The first line of the input gives the number of test cases, T . T test cases follow. Each test case begins with one line with two space-separated integers N and Q , denoting the number of elements in the initial array and the number of Alice's queries. Then, there is one line with N space-separated integers, denoting the elements of Alice's initial array. Finally, there are Q more lines with two space-separated integers each: L_i and R_i , the inclusive index bounds for the i -th query.

Output

For each test case, output one line with Case # x :, where x is the test case number (starting from 1). Then output Q more lines, each with one integer, representing the answers to the queries (in the order they were asked).

Limits

$1 \leq T \leq 10$.
 $1 \leq Q \leq 20$.
 $1 \leq$ each element of the initial array ≤ 100 .
 $1 \leq L_i \leq R_i \leq N*(N+1)/2$.

Small dataset

$1 \leq N \leq 10^3$.

Large dataset

$1 \leq N \leq 200000$.

Sample

Input	Output
1	Case #1:
5 5	1
5 4 3 2 1	45
1 1	105
1 10	26
1 15	48
3 8	
4 11	

In sample case #1, Alice's new array would be: [1, 2, 3, 3, 4, 5, 5, 6, 7, 9, 9, 10, 12, 14, 15].

All problem statements, input data and contest analyses are licensed under the [Creative Commons Attribution License](#).

© 2008-2017 Google [Google Home](#) - [Terms and Conditions](#) - [Privacy Policies and Principles](#)

Powered by



Google Cloud Platform