

CS2810 OOAIA: A7

Design Deadline: February 24 at 16:40 on Moodle
Code Deadline: February 26 at 23:45 on Hackerrank

Contest Link

<https://www.hackerrank.com/cs2810-ooaia-a7>

Objective

- To learn about Incremental sort and also use streams in C++.

Problem Statement

Create a class in C++ to represent Students. This class should also support the functionality to compare two Students based on the given constraints.

The class should have the following attributes.

- ID (Unique for a student)
- Scores in five subjects(No two students will have the same set of scores in all five subjects).
- Skill value (Unique for a student).
- Any other attributes of your choice.

The condition for comparing arbitrary students A and B is:

- $\text{Rank}(A) < \text{Rank}(B)$ if $M_1(A) > M_1(B)$
- $\text{Rank}(A) < \text{Rank}(B)$ if $M_i(A) = M_i(B)$ and $M_{i+1}(A) > M_{i+1}(B)$ for $i=1$ to 4 where $M_i(A)$ denotes marks of student A in subject i .

Rank is defined as the position of a student in the sorted order using the above condition. It is indexed from 1.

There can be 5 types of operations:

1. ADD: Add a student.
2. FIND: Find the difference in ranks between two students given their respective IDs.
3. COUNT: Given a student ID, find the number of students with absolute difference in skill values more than d . i.e. $\text{count}(|\text{val}_i - \text{val}_{id}| > d) \forall i$.
4. PRINT: Print the *current* list of student IDs in the order of their ranks.

As and when a new student is added it is expected that all of them are to be in sorted order i.e. positioning them in their appropriate rank.

You need to choose an appropriate data structure to perform operations efficiently.

Input Format

The operations format are as follows:

1. <ADD> <space> <studentID> <space> <Marks1> <space> <Marks2> <space> <Marks3> <space> <Marks4> <space> <Marks5> <space> <SkillValue>
2. <FIND> <space> <studentID_1> <space> <studentID_2>
3. <COUNT> <space> <studentID> <space> <d_value>
4. <PRINT>

Output Format

Output of every query should be printed on a separate line.

1. ADD: nothing to be printed. (no newline as well).
2. FIND: output is an integer.
3. COUNT: output is an integer.
4. PRINT: output is a space separated list of student IDs. Note that there is a trailing space in the end.

Constraints

Number of ADD operations $\leq 10^5$

Number of PRINT operations $\leq 10^2$

Number of FIND+COUNT operations $\leq 10^5$

Sample Testcase

```
ADD 1 28 15 72 79 53 880704511
ADD 2 9 27 78 89 81 133736473
ADD 3 94 2 98 99 43 102803388
PRINT
ADD 4 92 61 56 39 96 154547114
ADD 5 19 73 39 13 88 138195161
ADD 6 33 68 17 69 73 749942762
PRINT
FIND 2 3
COUNT 5 7006739
ADD 7 58 64 7 12 3 587319490
ADD 9 96 89 58 43 49 550217891
```

PRINT

Output

3 1 2

3 4 6 1 5 2

5

4

9 3 4 7 6 1 5 2

Design Submission Format

For the design submission on Moodle, please submit a **.tar.gz** file named as your **roll number**.