Geometric Sorting

locked

by cs16m044

Problem Submissions Leaderboard Discussions

Given n arbitrary coordinate points in 3-Dimensional space. You have to sort n points with respect to the given coordinates (reference point). Lowest value in your refernce point will have the highest priority, for example, 1 - highest priority and 3 - lowest priority.

For your Weekend assignment, you will require a modified "sorting" module. So, as part of your In-Lab assignment you will implement sorting of geometric points which will be used in your weekend assignment.

Input Format

- The first line contains an integer, n, denoting the number of coordinate points.
- Each of the n subsequent lines contains coordinate points. Format- x y z , there will be space between x, y, and z.
- Last line contains priority of the axis using which you have to sort the coordinates (reference points).

Constraints

n is an integer number. Coordinates are integer values. Each axis of reference point can lie between 0 to 3

Output Format

Print sorted coordinates in a new line.

Sample Input

3

2 1 9

0 3 1

1 2 3

1 0 0

Sample Output

0 3 1

1 2 3

2 1 9

Explanation

In sample input, 2 1 9 means 2 lies in x-axis, 1 lies in y-axis and 9 lies in z-axis, similarly the others. 1 0 0 is your reference point. Axis containing 0 in reference point will be ignored in the coordinate points. In reference point, except for the 0, repetition is not allowed, for example, 2 1 1 is not allowed.

If r1,r2,r3 is the reference point and two coordinates x,y,z and a,b,c are equal with respect to r1,r2,r3, then their relative ordering should be preserved. Example: Input

2

1 2 3

1 4 5

1 0 0

1 2 3 and 1 4 5 are equal with respect to the reference point 1 0 0.

Output:

1 2 3

1 4 5

Submissions: 67
Max Score: 60
Difficulty: Easy

Rate This Challenge:

More

Current Buffer (saved locally, editable) C++

```
#include <cmath>
#include <cstdio>
#include <vector>
#include <iostream>
#include <algorithm>
using namespace std;


int main() {
    /* Enter your code here. Read input from STDIN. Print output to STDOUT */
    return 0;
}

```

Line: 1 Col: 1

Upload Code as File Test against custom input Run Code Submit Code