CSE 4304-Data Structures Lab. Winter 2022-23
Date: 19 Sep 2023
Target Group: SWE B
Topic: Searching & Priority Queue

Instructions:

- Regardless of when you finish the tasks in the lab, you have to submit the solutions in the Google Classroom. The deadline will always be at 11.59 PM of the day in which the lab has taken place.
- Task naming format: <fullID>_<Task><Lab><Group>.c/cpp. Example: 170041034_T01L02A.cpp
- If you find any issues in the problem description/test cases, comment in the google classroom.
- If you find any test case that is tricky that I didn't include but others might forget to handle, please comment! I'll be happy to add.
- Use appropriate comments in your code. This will help you to easily recall the solution in the future.
- Obtained marks will vary based on the efficiency of the solution.
- Do not use the <bits/stdc++.h> library.

Task:1

Implementing the basic operations of a **Heap** using an Array.
Implement the following functionalities:

1. int **Heap_Minimim**(int heap[]): Returns the minimum value.
2. int **Heap_extract_min**(int heap[]): Removes the minimum value and returns it.
3. **Min_heap_insert**(int value, int heap[]): Inserts the 'value' into the heap and makes necessary arrangements.
4. **Heap_decrease_key**(int p, int k, int heap[]): Decreases the value at p-th position by an amount of k and makes necessary changes. You need to go up for keeping the min heap property.
5. **Heap_increase_key**(int key, int amount, int heap[]): Increases the value at p-th position by an amount of k and makes necessary changes. You need to go down for keeping the min heap property.

Input
First line of input will contain a set of numbers. Show the corresponding min-heap for that.
After that the input will be like 'function_id necessary_params (if any)'. Show the output and 'state of the heap' after each function call.

Input	**Output**
70 90 60 190 80 170 110 20 50 30 130 -1	20 30 60 50 70 170 110 190 90 80 130
1	20 20 30 60 50 70 170 110 190 90 80 130
2	20 30 50 60 90 70 170 110 190 130 80
1	30 30 50 60 90 70 170 110 190 130 80
3 45	30 45 60 90 50 170 110 190 130 80 70
4 4 65	25 30 60 45 50 170 110 190 130 80 70
2	25 30 45 60 70 50 170 110 190 130 80
5 1 170	45 50 60 70 80 170 110 190 130 200
3 47	45 47 60 70 50 170 110 190 130 200 80

Task:2

Mark loves cookies. He wants the sweetness of all his cookies to be greater than the value of **K**. To do this, Mark repeatedly mixes two cookies with the least sweetness. He creates a special combined cookie with:
Sweetness = (1 x Least sweet cookie + 2 x *2nd least sweet cookie*).
He repeats this procedure until all the cookies have a sweetness ≥ **K**.
You are given Mark's cookies. Print the number of operations required to give the cookies a sweetness ≥ **K** Print **-1** if this isn't possible.

You must use Heap/Priority queue. You can use STL.

Input format
The first line consists of integers **N** representing the number of cookies, and **k**- the minimum required sweetness, separated by a space.
The next line contains **N** integers describing the array ***A*** where A_i is the sweetness of the i^{th} cookie in Mark's collection.

Output format
Output the number of operations that are needed to increase the cookie's sweetness ≥ **K**
Output **-1** if this isn't possible.

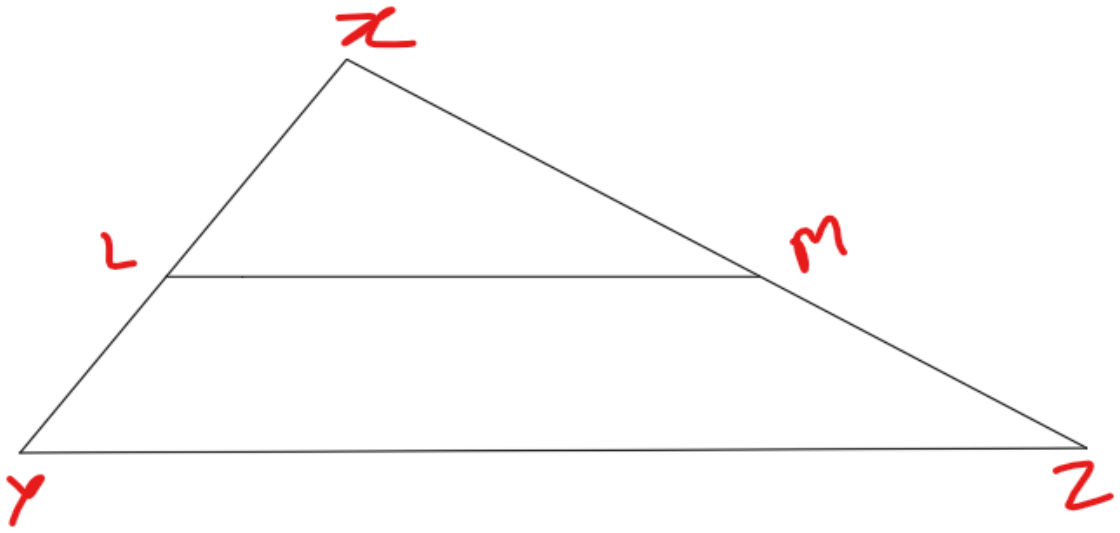
Sample Input	**Sample Output**
6 7 12 9 1 3 10 2	2

Explanation
Combine the first two cookies to create a cookie with *sweetness* = 1 x 1 + 2 x 2 = 5
After this operation, the cookies are (3, 5, 9, 10, 12)
Then, combine cookies with sweetness 3 and sweetness 5, to create a cookie with resulting *sweetness* = 1 x 3 + 2 x 5 = 13
Now, the cookies are (9, 10, 12, 13).
All the cookies have a sweetness >= 7

Thus, **2** operations are required to increase the sweetness.

Task:3

In the given geometrical figure, LM is parallel to YZ.



Now you are provided with the lengths of XY, XZ, YZ and the ratio of area XLM and LMZY. Find the length of XL.

You have to use binary search.

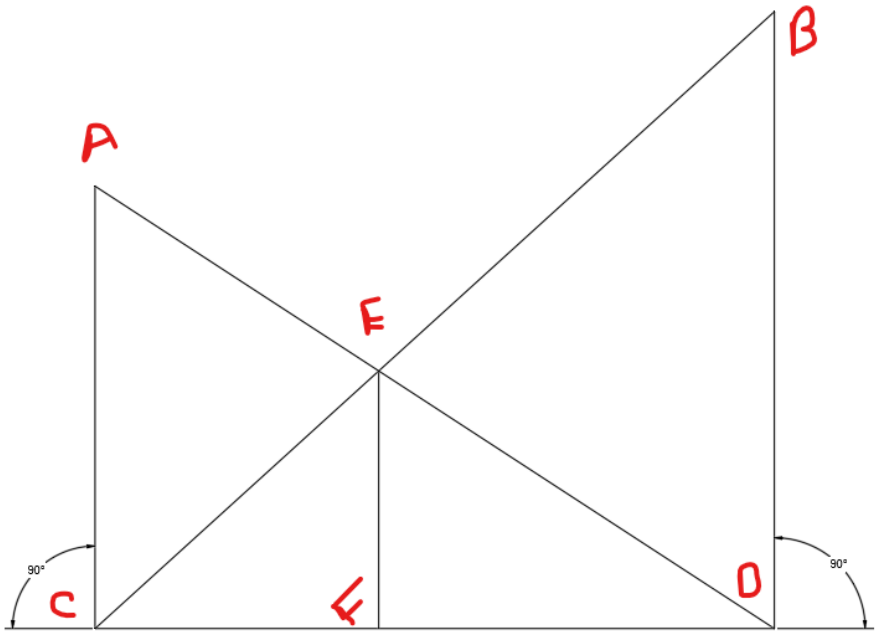
Input
Input starts with an integer T which is the number of test case.
Each of the following T lines will have four real numbers denoting XY, XZ, YZ and the ratio of area (XLM / LMZY).

Output:
For each case print the case number and the value XL. The precision error should be less than 0.000001.

Input	Output
4 8.134 9.098 7.123 5.10 7 8 9 10 100 100 100 2 10 12 14 1	Case 1: 7.437454786 Case 2: 6.6742381247 Case 3: 81.6496580 Case 4: 7.07106781

Task:4

In the following geometry, the length of AD, BC and EF is provided two you. You have to



find the length of the CD.

Use binary search on the condition that if CD increases EF gets smaller, if CD decreases EF gets larger.

Input:
Input starts with an integer T which is the number of test case.
Each of the following T lines will have 3 real numbers denoting the value of AD, BC and EF.

Output:
For each case print the case number and the value CD. The precision error should be less than 0.000001.

Input	Output
4 10 10 1 30 40 10 10 10 3 12.619429 8.163332 3	Case 1: 9.797958971 Case 2: 26.0328775442 Case 3: 8 Case 4: 6.99999923