

Question 1**MARKS : 20**

Your friend is a chief architect, who is working on building a skyscraper, in Mumbai. The construction is in such a way that the floors will be constructed in other factories and they will be assembled. All the constructed sizes will be **distinct**.

The skyscraper needs to be constructed in N days with the following conditions :

- a) Every day a floor is constructed in a separate factory of **distinct** size.
- b) The floor with the larger size must be placed at the bottom of the building.
- c) The floor with the smaller size must be placed at the top of the building.

Note: A floor cannot be assembled in the building until all floors larger in size are placed.
As a friend he wants you to build a small program that will help him analyze the construction process, to avoid manual work and errors.

Input

First Line: contains the total floors N in the building

Second Line onwards: Contains N inputs in which the ith integer denotes the size of the floor that will be given to architect by factories on the ith day.

Output

You are required to print N lines. Print the size of the floor in descending order, which can be assembled.

If no floor can be assembled on the ith day, leave the line empty.

Sample Test Cases**TestCase 1**

enter the total no of floors in the building

3

enter the floor size given on day : 1

3

enter the floor size given on day : 2

1

enter the floor size given on day : 3

2

The order of construction is as follows

Day: 1

3

Day: 2

Day: 3

2 1

Explanation

The largest floor size is available on day 1 itself, hence it will be assembled on day 1.

On the second day floor of size 1 is available and cannot be assembled as the floor of size 2 is not available.

On the third day floor of size 2, is available and accordingly, the remaining floor (2,1) will assemble in the order 2,1

TestCase 2

enter the total no of floors in the building

5

enter the floor size given on day : 1

4

enter the floor size given on day : 2

5

enter the floor size given on day : 3

1

enter the floor size given on day : 4

2

enter the floor size given on day : 5

3

The order of construction is as follows

Day: 1

Day: 2

5 4

Day: 3

Day: 4

Day: 5

3 2 1

Explanation

On the first day, the floor of size 4 is given. But it cannot be assembled as the floor of size 5 is still remaining.

On the second day, the floor of size 5 is given so now the floor of sizes 5 and 4 can be assembled

On the third and fourth days, floors cannot be assembled as the size of 3 needs to be given yet. Therefore, these lines are empty.

On the fifth day, all the floors of sizes 3, 2, and 1 can be placed.

Test Case 3

enter the total no of floors in the building

10

enter the floor size given on day : 1

1

enter the floor size given on day : 2

2

enter the floor size given on day : 3

3

enter the floor size given on day : 4

8

enter the floor size given on day : 5

9

enter the floor size given on day : 6

10

enter the floor size given on day : 7

4

enter the floor size given on day : 8

5

enter the floor size given on day : 9

7

enter the floor size given on day : 10
6

The order of construction is as follows

Day: 1

Day: 2

Day: 3

Day: 4

Day: 5

Day: 6

10 9 8

Day: 7

Day: 8

Day: 9

7

Day: 10

6 5 4 3 2 1

Instructions :

- 1) You can use any inbuilt java function/s to implement the above functionalities
- 2) You can choose any DataStructure(Stack, Queue, LinkedList) to implement the above functionality.



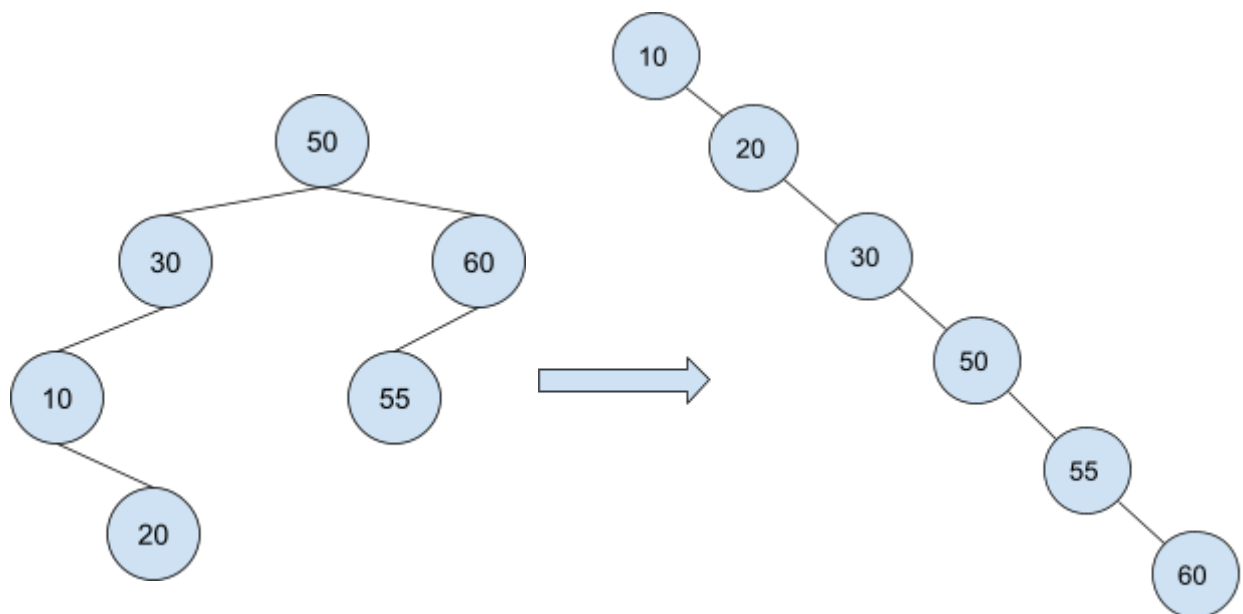
Question 2**MARKS: 30**

You are working in an MNC, which manages the Transactions, where only BST is used as a Data Structure. The company stores all the data of transactions in BST such that the tree is always a complete BST.

A new business requirement has arrived where the BST should not contain any left node.

You are required to modify the existing BST and display the node values present in ascending order.

Hint: (Convert the Binary Search Tree into a Skewed Tree).

**Output**

10 20 30 50 55 60

Instructions

You can hardcode the tree as follows

```
Main tree = new Main();
tree.node = new Node(50);
```

```
tree.node.left = new Node(30);  
tree.node.right = new Node(60);  
tree.node.left.left = new Node(10);  
tree.node.left.left.right = new Node(20);  
tree.node.right.left= new Node(55);
```

Learning Objectives:

1. Work and enhance logical skills to write efficient scripts.
2. Apply the concepts of Binary Trees and any required DSA concepts to solve the problems.