<u>Task 1:</u>

Disneyland has built their own airport. The traffic of transportation is obviously very high. But unfortunately they have only one runway. So they have decided to create a '***Runway reservation system***' for their only runway which will take the reservation of any transport desired to use the runway. However, before taking the reservation, the system checks if there is already a reservation within 3 minutes range of any existing reservation(e.g. If there is a reservation in the kth minute, (it won't take any reservation in k-1, k-2, k-3, k+1, k+2, k+3th minutes) In such a case, the system will not take the reservation.

Your task is to help them build the system using Binary Search Trees(BST). (Take reservations until the user gives '-1' as input.)

For every reservation, print the existing reservations in sorted manner.

Sample Input	**Sample Output**
50 75 53 25 60 29 45 42 28 10 -1	50 50 75 50 75 (Reservation failed) 25 50 75 25 50 60 75 25 29 50 60 75 25 29 45 50 60 75 25 29 45 50 60 75 (Reservation failed) 25 29 45 50 60 75 (Reservation failed) 10 25 29 45 50 60 75

<u>Task 2:</u>

Perform ‘Level-order traversal’ on the tree built in Task-1.

Sample Input	**Sample output**
50 75 25 29 45 60 10 -1	50 25 75 10 29 60 45

<u>Explanation:</u>

1st level - 50
2nd level - 25 75
3rd level - 10 29 60
4th level - 45

<u>Hint:</u> You might need to use a stack/queue to keep track of the nodes to be visited. (E.g queue<node*> q)

<u>Task 3</u>

A great battle is going on between Mike and his enemy team. The General of the enemy team is very clever. He(the General) decided to arrange his soldiers in a strategic way. Firstly, he went to the battlefield and told the soldiers to come one after another. Each soldier(including the General) has his own power.

Whenever a new soldier comes, the General compares the power of the new soldier with his own power. If it is less or equal, that soldier goes to the left of him(the General), otherwise right side. However, the General follows this unique property for every internal soldier(that means, if there are more than one soldier, a new soldier will need to satisfy the property for each of the soldiers including the general).

Now, Mike wants to attack from a random end. Your task is to find out what is the maximum number of soldiers that Mike might need to beat to reach the General of the enemy team.

Each test case starts with the power of the General followed by a number of soldiers unlit you get -1.

Sample input	**Sample output**
50 25 29 75 10 60 45 -1	3
50 60 70 10 55 -1	2
100 120 110 55 35 45 130 49 -1	4

<u>Explanation:</u>

For first case, if the soldiers are organized properly, it will look like

```
        50
       /     \
    25        75
  /     \     /
10      29  60
           \
            45
```

So in the worst case, if mike start killing soldier with power = '45', he'll need to kill 45, 29, 25 before reaching the General with power = 50

Task 4

Coming back to the ‘Runway reservation problem’, now Disneyland airway wants you with a new requirement. They want to introduce a new feature in their system, which says, any Transport owner who got a reservation at a certain slot can now make a query to the system to know the timestamp of the reservation which will be served before & after his transport.

One of their employees proposed a solution that traverses the tree in In-order fashion and then finds the timestamps that are less than the query. They are not happy with this ***O(n)*** solution. They want you to solve this problem in ***O(height)*** time.

First line of input will give you the number of queries.
Each query gives you the timestamp of a certain reservation. Your task is to show the timestamp of the reservation which will be served before & after that reservation. If there is no reservation after certain reservation, print NULL. Same if no reservation exists before a certain reservation.
If the query matches with none of the existing reservations, output will be ‘reservation not found!’.

Sample input	**Sample output**
(current reservations) 50 75 25 29 45 60 10 -1	
5	
25	10 29
45	29 50
75	60 NULL
10	NULL 25
50	45 60
55	Reservation not found!

Task 5:

‘Runway reservation system’ has given a new requirement. They want to introduce a feature which will allow any transport owner to make a new query, which is, if any transport-owner gives the timestamp as input, the system will tell ‘How many reservations’ are in the system before it.

One of their employees proposed a solution that traverses the tree in In-order fashion and then finds the timestamps that are less than the query. They are not happy with this ***O(n)*** solution.
They want you to solve this problem in ***O(height)*** time.
Your task is to fulfill their requirement.

First line of input will give you the number of queries.
Each query gives you the timestamp of a certain reservation. Your task is to find the number for reservation before that timestamp.

Sample input	**Sample output**
(current reservations) 50 75 25 29 45 60 10 -1	
5	
45	3
75	6
50	4
10	0
29	2

Explanation:

45 has 3 before it (10 25 29)
75 has 6 before it (10 25 29 45 50 60)
50 has 4 before it (10 25 29 45)

[Hint: You might need to use a new attribute for each node called ‘subtree-size’. Start traversing from the root and try to use that subtree-sizes wisely. This will lead you to a solution of O(height).]

Task 6

Suppose a set of numbers are stored using a Binary Search Tree(BST). Two numbers **x** and **y** are given as input(x<y). Your task is to find the maximum number that will be encountered from the path going from x….y .

[**O(n)** solution will not be accepted.]

Sample input	**Sample output**
(current values) 50 75 25 29 45 60 10 80 -1	

8
10 50
25 45
60 80
25 60
12 25
10 60
50 60
75 80
	
	50
	45
	80
	75
	25
	75
	75
	80

Explanation:

```
         50
      /      \
     25         75
  /     \    /   \
 10       29  60    80
  \        \
   12       45
```

Now if the value of (x,y) = (10,50) the path to reach from 10...50 contains the nodes with values {10,25,50}. We need to pick the maximum value amongst them, which is 50.

If (x,y) = (25,60) the nodes encountered in the path from 25...60 are {25, 50, 75, 60}. The maximum value amongst them is 75.

Task 7:

‘Runway reservation system’ has given their final requirement. Now they want your help to introduce a new feature which will show the updated list of reservations after a certain transport has used the runway(that transport will be taken out of the set of reservations).

First line of input will give you the number of queries.
Each query gives you the timestamp of a transport which has used the runway. Your job is to remove the reservation and show the updated set using level-order traversal.

Sample input	**Sample output (level-order traversal)**
(current reservations) 50 75 25 29 45 60 90 10 80 100 84 88 -1	
6 29 25 75 45 50 80	50 25 75 10 29 60 90 45 80 100 88 ***(initial)*** 50 25 75 10 45 60 90 80 100 84 88 50 45 75 10 60 90 80 100 84 88 50 45 80 10 60 90 84 100 88 50 10 80 60 90 84 100 88 60 10 80 90 84 100 88 60 10 84 90 88 100