WEEK-9	COST ANALYSIS
LAB	3-8 OCT 22

Q1. It has been proven that the *average* search cost for an arbitrary item in a binary search tree of N
random items is O(log N). You are to verify this result experimentally. Write a program that prompts the user for the number of items to be stored (N) and the number of trees to generate (T). Then, it should repeatedly (T times) store N random numbers in a binary search tree and compute the average cost of searching that tree.

Your program should display the average of these costs over all of the constructed trees. In addition, it should display the average height of the trees. For example,
Number of values to be stored: 1000
Number of trees to generate: 100
Generating 100 trees with 1000 random values:
average cost = 11.9146
average height = 21.76
Run your program from part 2 for various values of N, using T = 1,000. Report the average height and cost of searching the trees you constructed. Do your statistics support the claims that the average height and cost of searching a randomly constructed binary search tree are both O(log N)? Justify your answer.

number of values (N)	log2(N+1)	average cost	average height
N = 1,000			
N = 2,000			
N = 3,000			
N = 4,000			

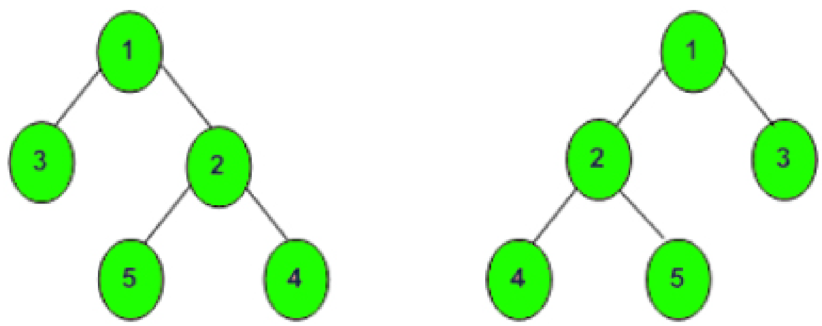
WEEK-	
	3-8 OCT 22

Q1 Given a Binary search tree. Your task is to complete the function which will return the Kth largest element without doing any modification in Binary Search Tree.

```
Input:
      4
    /   \
   2     9
k = 2
Output: 4
```

```
Input:
       9
        \
         10
K = 1
Output: 10
```

Q2. Write a program to convert a binary tree to mirror tree



Mirror Trees

Q3 Given two BSTs, return elements of both BSTs in sorted form.

```
Input:
BST1:
       5
     /   \
    3     6
   / \
  2   4
BST2:
        2
      /   \
     1     3
            \
             7
            /
           6
Output: 1 2 2 3 3 4 5 6 6 7
Explanation:
After merging and sorting the
two BST we get 1 2 2 3 3 4 5 6 6 7.
```