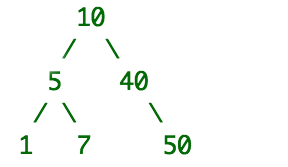
Have the function BinarySearchTreeLCA(**strArr**) take the array of strings stored in **strArr**, which will contain 3 elements: the first element will be a binary search tree with all unique values in a [preorder traversal](https://en.wikipedia.org/wiki/Tree_traversal" \l "Pre-order" \t "https://coderbyte.com/results/_blank) array, the second and third elements will be two different values, and your goal is to find the [lowest common ancestor](https://en.wikipedia.org/wiki/Lowest_common_ancestor" \t "https://coderbyte.com/results/_blank) of these two values. For example: if**strArr** is ["[10, 5, 1, 7, 40, 50]", "1", "7"] then this tree looks like the following:   
  
   
  
For the input above, your program should return **5** because that is the value of the node that is the LCA of the two nodes with values 1 and 7. You can assume the two nodes you are searching for in the tree will exist somewhere in the tree.   
  
Use the **Parameter Testing** feature in the box below to test your code with different arguments