**Question:**

Write a recursive function **find\_elements\_leq()** that takes a BST and a target number, and returns a **string** with all BST elements **less than or equal** to the target. Use **preorder traversa**l to populate the string.  
  
**Note:** You cannot use any loops or global variables. You may use any number of helper functions as needed. Assume the TreeNode class is provided, where each node contains an integer value in the element attribute and pointers to the left and right child nodes.

| **Sample Input** | **Sample Output** |
| --- | --- |
| 10  / \  5 15  / \  2 7 | **10 5 2 7** |
| Target = 10 |
| # Python Function Signature  def find\_elements\_leq(root, target, result=None)  # Python Function Call  print(find\_elements\_leq(root, target)) | |
| //Java Method Signature  public static String find\_elements\_leq(TreeNode root, int target)  //Java Method Call  System.out.println(find\_elements\_leq(root, target)); | |

**Question:**  
Write a recursive function **find\_elements\_geq()** that takes a BST and a target number, and returns a **string** with all BST elements **greater than or equa**l to the target. Use **preorder traversa**l to populate the string.  
  
**Note:** You cannot use any loops or global variables. You may use any number of helper functions as needed. Assume the TreeNode class is provided, where each node contains an integer value in the element attribute and pointers to the left and right child nodes.

| **Sample Input** | **Sample Output** |
| --- | --- |
| 10  / \  6 16  \ /  8 13 | **10 16 13** |
| Target = 10 |
| # Python Function Signature  def find\_elements\_leq(root, target, result=None)  # Python Function Call  print(find\_elements\_leq(root, target)) | |
| //Java Method Signature  public static String find\_elements\_leq(TreeNode root, int target)  //Java Method Call  System.out.println(find\_elements\_leq(root, target)); | |