1. Write a function that receives a reference to the root of a binary search tree and an integer \( k \) and prints all the items in the tree that are less than \( k \).

2. Write a function that receives a reference to the root of a binary search tree and prints all the items in the tree that are stored in leaf nodes.

3. Write a function that receives a reference to the root of a binary search tree and an integer \( d \) and returns the largest item in the tree that is stored at depth \( d \), or \(-\text{math.inf}\) if the tree has no items with depth \( d \).

4. Write a function that receives a reference to the root of a binary search tree and an integer \( d \) and prints all the items in the tree that are stored at depth \( d \).

5. Write a function that receives a reference to the root of a binary search tree and an integer \( k \) and returns the depth of the node that contains \( k \), or -1 if \( k \) if not in the tree.

6. Write a function that receives a reference to the root of a binary search tree and returns a sorted list containing the elements in the tree.

7. Write a function that receives a sorted list \( L \) and builds and returns a balanced (as much as possible) binary search tree containing the elements of \( L \).

8. Write a function that receives a reference to the root of a binary search tree and prints the items in the tree ordered by depth.