Write a C++ class implementing binary search tree that supports the following operations:

1. [10 points] insert (x) -- insert an element x (if not present) into the BST. If x is present, throw an exception.
2. [10 points] search(x) -- search an element x, if found return its reference, otherwise return NULL .
3. [20 points] delete(x) -- delete an element x, if the element x is not present, throw an exception.
4. [20 points] reverseInorder() -- returns a singly linked list containing the elements of the BST in max to min order. You should not use any extra stack and use threading for non-recursive implementation.
5. [10 points] successor(ptr) -- returns the key value of the node which is the inorder successor of the x, where x is the key value of the node pointed by ptr.
6. [20 points] split(k) -- split the BST into two BSTs T1 and T2, where keys(T1) <= k and keys(T2) > k. Note that, k may / may not be an element of the tree. Your code should run in O(h) time, where h is the height of the tree. Print the inorder of both the BSTs T1 and T2 using recursive/non-recursive implementation within this function.
7. [20 points] allElementsBetween(k1, k2) -- returns a singly linked list (write your own class) that contains all the elements (k) between k1 and k2, i.e., k1 <= k <= k2. Your code should run in O(h + N) time, where N is the number of elements that appears between k1 and k2 in the BST.
8. [20 points] kthElement(k) -- finds the k-th largest element in the BST and prints the key value. Your code should run in O(h) time.
9. [20 points] printTree() -- Your program should print the BST (in a tree like structure) [use graphviz]

Points to note:

1. Use gcc/g++ compiler and same code should run in both linux and windows (dev C++)
2. Code should be sufficiently commented.
3. Variable names should be reasonable.
4. Prepare a document (PDF) clearly stating your logic for implementing above mentioned functions, test cases and their outputs (printed as a tree).
5. Note that, we will use automated tool to check plagiarism in the source code and penalize heavily in case of unfair means and copying others' codes.

6. Maximum points possible = 150 points.

Submission Guidelines:

1. Submission deadline: Aug 21, 2021 11:59pm. Deadlines should be strictly followed. Late submissions will not be considered for evaluation.
2. Prepare a zip file containing source files (.cpp files), the PDF document and a readme file that clearly states how to run the code.
3. Name the zip file as BST_<rollno>.zip and send it to "cs513submission@gmail.com"