CMSC 204: EXERCISE 2

For this exercise, you will be working with BSTs. The output required for Exercise 2 is a compiling C# code bundled inside **LastnameExer2.zip**.

IMPORTANT NOTES:

1. You are to follow the naming convention specified. You will be given a deduction if you fail to do so.
2. Make sure that you document your code (include your name, student number, the date and a brief description of the program in comments at the start of the code). Also put in the necessary comments that will help you (and me) understand your code.
3. To ensure the validity and integrity of your work, your programming exercise should come with a screencast of how you coded your work. You can use a screencasting freeware called Jing for Mac and Windows (http://www.techsmith.com/jing.html) or any alternatives. The screencast of your work should be at most 5 minutes (deductions if it exceeds the time), which will contain a coding demonstration of your solution. You don't have to capture the whole programming process. You can just highlight the important parts and explain how you did your assignment. The final section of the video should feature the finished product of your work and a short demo on how it works. You should be the one doing the voice-over for the explanation. The explanation can be in Filipino or English. Name the video as **LastnameExer2.** Note that the upload limit for the portal is 64MB. If your video is more than that, you can upload it to Youtube as "unlisted" and then you can include the link in your submission. Note that only one video is needed as an output for this exercise.

THE EXERCISE

Using the **LINKED LIST IMPLEMENTATION***:

1. Display the following menu:

```
[1] Insert node to binary tree
[2] Delete node from binary tree
[3] Minimum
[4] Maximum
[5] Successor
[6] Predecessor
[7] Search
[8] Print BST
```

2. Start off with an empty tree.
3. When the user chooses [1], the program prompts the user to enter a value. After searching for its proper position, the node is then inserted into the BST.
4. When the user chooses [2], the program asks the user what value to delete. The value is deleted if it is found in the BST. An error is shown if no value exists.
5. Choosing [3] displays the minimum value in the BST.
6. Choosing [4] displays the maximum.
7. Choosing [5] would display the successor of a node of which the value has been specified by the user.
8. Choosing [6] displays the predecessor of the node whose value has been specified by the user.
9. Choosing [7] searches for the value specified by the user. Display a message if the value is not in the BST.
10. Choosing [8] prints out the contents of the BST in increasing order.

* IMPORTANT: Create your own linked list. **Do not make use of the built in lists and queues in C#.**

FOR THE BRAVE: You can choose to implement the exercise using a Windows interface and accept input via forms or whatnot instead of a* ***console window*. As long as you can implement the functionalities specified, Windows based* *applications will be accepted. Bonus points will be given if you are able to deliver more than what is required.*