Homework 4

100 Points

Binary Search Trees - ADT

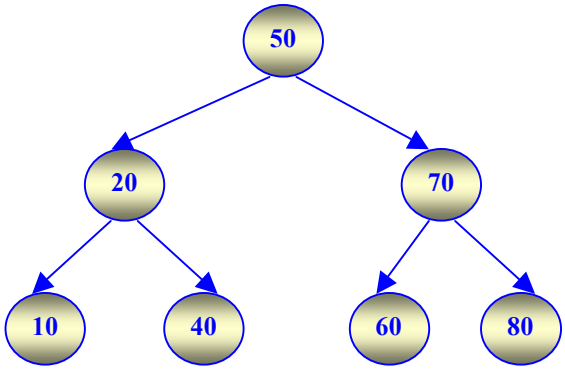
Write a menu-driven program that implements a simple BST-based database. The program reads data from a text file and inserts them into a BST. Once the trees has been built, present the user with a menu:

D – Depth-First Traversals: inorder, preorder, postorder // recursive
I – Iterative Depth-First Traversals: inorder, preorder, postorder (use a stack) // EC – 1Point
(Use the CIS22C stack library, no STL please)
B – Tree Breadth-First Traversals: Print by level // iterative
(Use the CIS22C queue library, no STL please)
S – Search by the primary key (student ID – unique key) // iterative
M – Find the smallest // recursive
X – Find the largest // recursive
T – Print the BST as an indented list (show level numbers)// see below // recursive
A – This is a hidden option. When chosen, display the name of the developer.
H – Help – to show the menu, and
E – Exit.

Provide at least one test case for each option, except for search, for which provide more test cases. Send the output to the screen. When done, copy and paste it to `BST_Output.txt`.
Input File: `gpa.txt` (see next page).

Grading

1. Build BST	– 10
2. Insert BST (recursive)	– 10
3. Recursive Depth-First Traversals	– 10
4. Breadth-First Traversal	– 10
5. Search BST	– 10
6. Find smallest	– 10
7. Find largest	– 10
8. Print Tree	– 10
9. Destroy Tree	– 5
10. main(), Options A, H, E, etc.	– 10
11. Self assessment	– 5



Print tree as an indented list

```
1. 50
      2. 70
            3. 80
            3. 60
      2. 20
            3. 40
            3. 10
```

To test your program create an input file using the data shown below and name it **`gpa.txt`**:

```
3800 Lee, Victor; 2.8
3000 Brown, Joanne; 4.0
3500 Marcus, John; 3.2
1900 Walljasper, Bryan; 3.9
4300 Trapp, Amanda Elizabeth; 4.0
2100 Andrews, Dennis; 2.9
3900 Brown, Joanne; 3.5
5000 Lee, Lucy Marie; 3.4
2800 Brown, Joanne; 2.6
4200 Brown, Joanne; 4.0
3300 Lee, Victor; 3.6
1500 Marcus, John; 4.0
2500 Lee, Mary; 3.8
```

The Student object has three data members: id (the unique key), name, and gpa.

Requirement: BST ADT
As en entry-level programmer you have to be able to read, understand existing code and update it (add new features). One of this assignment's goals is to read code:

BinaryNode.h,
BinaryTree.h.cpp,
BinarySearchTree.h, and
main.cpp,

compile and run the program, understand it, and change it as required.