Question 1:   [15 Marks]

Create a class called **Team**. This class should have the following properties:
a. name : string
b. initial : string
c. numberofgoals : int

a. The class should have a print function that prints all information about the teams. Overload the comparison operators "<", ">", "<=", ">=", "==" and "!=" based on the teams' number of goals. In the driver file, create two Team objects and then compare the two teams using the comparison operators you overloaded earlier and print output like *"Brazil **has scored more goals than** Argentina."*

b. You need to make a point table for the teams based on the number of goals scored. For this, create a SortedList class that works only for the Team objects. In the driver file, create a few Team objects, insert them into a SortedList, and finally, print the list. *[N.B: Add an empty constructor on Team class while doing this otherwise, it may cause errors]*

c. Implement the delete function of the sorted list with recursive binary search.

Question 2:   [10 Marks]

Use your NSU ID digits in the following stacks.

| | batch | id | dept |
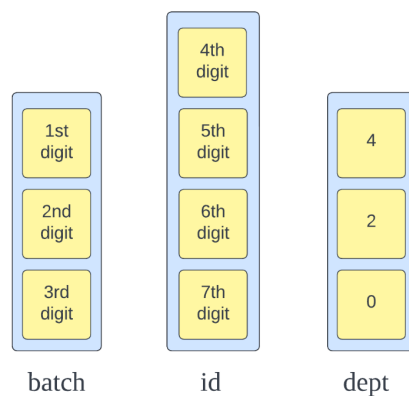
Implement the stack data structure. The stack class must have two added features named: **mergeStack** and **reverseStack.**

The merge stack will join two stacks into one stack and return the merged stack.
The reverse stack will return a stack that is the reverse of the given stack.

Following the order of the diagram above, create the three stacks - batch, id, and dept. Merge the batch and id stacks so that it can easily print your first seven digits of ID. After printing the first seven digits, reverse the dept stack to represent your actual department (0 4 2). Now merge the reversed dept stack with the previous stack and print your complete NSU ID.

Question 3:   [5 Marks]

Create a program that can determine whether the user input is palindrome or not. You must implement it using the **Queue** data structure.

Question 4:   [10 Marks]

Modify the binary search tree and add two functions. The first one should count the number of leaf nodes, and the second one should calculate the height of the tree.
In the driver file, create an integer binary search tree, insert 10 integers in the tree and finally print the number of leaf nodes and the height of the tree.