



# CSE225: Data Structures and Algorithm

## Course Assignment

Section: 8, Fall '22

Total Marks: 40

North South University

### Question 1:

[15 Marks]

Create a class called **Goalkeeper**. This class should have the following properties:

- a. name : string
- b. country: string
- c. numberOfSaves : int

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

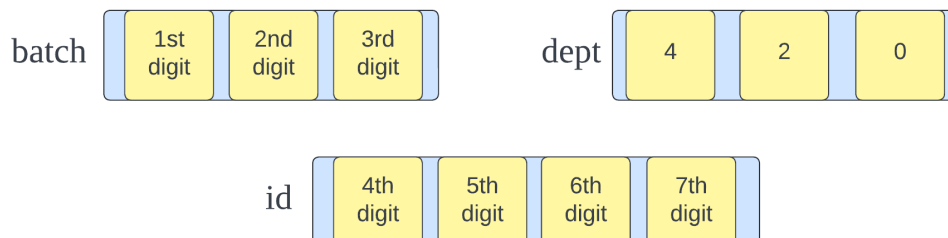
b. You need to make a point table for the goalkeepers based on the number of saves accomplished. For this, create a SortedList class that works only for the goalkeeper objects. In the driver file, create a few goalkeeper objects, insert them into a SortedList, and finally, print the list. [N.B: Add an empty constructor on the goalkeeper 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 queues.



Implement the queue data structure. The queue class must have two added features named: **mergeQueue** and **reverseQueue**.

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

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