NYU, Tandon School of Engineering
Bridge to Computer Science Program — Summer 2021

Third Midterm Exam

- There are 100 points total.
- Note that there are longer programming problems at the end. Be sure to allow enough time for these.
- We supplied you with a file, named ‘solutions.txt’, where you should type all your answers in.
- For editing this file, you are allowed to use compilers such as Visual Studio, XCODE and CLion
- You may use 2 scratch papers.
- Calculators are not allowed.
- This is a closed-book exam. No additional resourced are allowed.
- Pay special attention to the style of your code. Indent your code correctly, choose meaningful names for your variables, define constants where needed, choose most suitable control statements, etc.
- In all questions you may assume that the users enter inputs as they are asked. For example, if the program expects a positive integer, you may assume that users will enter positive integers.
- No need to document your code in this exam, but you may add comments if you think they are needed for clarity.
- Read every question completely before answering it.
- When finished, please submit on Brightspace, Gradescope AND email to dkatz@nyu.edu

1. (3 pts) Which of the following would be the appropriate function header for the overloaded post-increment operator for a Linked List of integers (assume you will want to increment was value in the list by one).
    a. LList& LList::operator++(LListNode ptr)
    b. LList LList::operator++(int)
    c. LListNode& LListNode::operator++(int)
    d. LListNode LListNode::operator++(int)

2. (3 pts) When defining a class which will contain a function that cannot be defined in the base class, but must be defined in derived classes, we should define the function as

    a. Overloaded
    b. Virtual
    c. Pure-Virtual
    d. Abstract

3. (3 pts)Which of the following lines of code would evaluate to false in a binary search tree

    a. ptr->data > ptr->left->data
    b. ptr->left->data > ptr->left->left->data
    c. ptr->right->data > ptr->right->left->data
    d. ptr->left-data > ptr->right->data

4. (3 pts) Convert the math expression "`2+3*4*(6-5)`" to post fix form.

5. (3 pts) Evaluate the post fix expression "`3 2 6 + * 2 /`" to a value.

6. (15 pts) Explain, in English not code, how you would determine the median element in a Linked List.

7. (20 pts) Normally, a queue is FIFO however it is theoretically possible to periodically, upon calling a function, "sort" the items that are in the queue so that the minimum value is at the front of the queue and the maximum at the end (all others are in normally "sorted" order within the queue). Please define the "sortQueue" function which will take a queue of unknown data type (you can assume the type has the less-than operator defined) and will sort the elements in the queue. This function will be a NON-member of the class! You may make use of any data structure in STL or our review (LList, Stack, Queue, Tress, BST, AVL, etc) and do not have to define these classes, simply use them if you'd like.

8. (15 pts) Due to new COVID restrictions in place at NYU, in order to come onto campus now, you must have completed the "Daily Screener." The userIDs (for example, dkb238) for anyone who has completed the "Daily Screener" is stored in a file called "daily.txt" (you can trust that this file exists, no need to test for it's existence). You are being asked to write a function (fillFromFile) to read this file into a data structure of your choice so that it can be searched efficiently by the ID card readers at the door to each building. You may choose any datatype for storage (either STL or one we created; no need to provide this datatype, we will assume a form similar to what we used in class). Read the file, load the values into your datatype and return the datatype from this function.

9. (20 pts) For efficiency, we're adding a "size" parameter to each node of a Binary Search Tree, similar to the way we added a height parameter to each node for the AVL tree. This size is stored in the node and not calculated so accessing it is constant time. We would like you to write a function which will determine the "Nth" element in order. That is, if the in-order traversal of the tree is "2,3,5,6,8,9" the 3rd element is 5, the 6th element is 9, etc. You function should be a member of the BST class and should have the form "T BST<T>::findNth(int n)"

10. (15 pts) A PhoneBook is a datatype that contains a set of PhoneEntry items stored as a vector inside the data type. The PhoneEntry data type is defined for you, you do not have to create it. It contains a function call "getName" which gets the name (a string) of the person for whom the entry references, you can assume that no two people have the same name so this becomes a unique value.

    You are tasked with writing the PhoneBook class. The class should store the vector of PhoneEntry objects and must maintain uniqueness. You must have functions to allow for

    - A size function
    - An operator to access an entry given its index in the vector.
    - adding a new PhoneEntry object (returns a Boolean true if successful or false if the name is a duplicate)
    - An overloaded operator to allow for adding two PhoneBook objects together (duplicates cannot occur)
    - An overloaded operator to allow for removing entries from a phone book (the result would be those entries that exist in the left-hand side which do NOT appear in the right-hand side)
    - An output operator which will output all of the names of the PhoneEntry items in the vector, one per line.