

**CSC 228 Data Structures and Algorithms, Fall 2017**  
**Instructor: Dr. Natarajan Meghanathan**

**Quiz 3 (Take Home): Due on Oct. 9 @ 1 PM.**

**Total Points: 50**

You are given the startup code for hash tables for both the following questions. Your task is to extend the main function and/or add/modify the member functions of the Hashtable and/or the List class to accomplish the required.

Q1 - 25 pts) Input two sequences of integers (as Strings and then use String Tokenizer to extract the individual integers) and use a hash table to determine whether the relationship between the second integer sequence and the first integer sequence is one of the following:

- (i) the second integer sequence is a *proper subset* of the first integer sequence
- (ii) the second integer sequence is a *permutation* of the first integer sequence
- (iii) the second integer sequence is neither a permutation nor a proper subset of the first integer sequence.

Note that an integer sequence S2 is a "proper subset" of integer sequence S1 if every element of S2 is in S1, but S1 has at least one element that is not in S2.

You are given the code for the Hash table permutation check that we discussed in class. Modify this code to identify the relationship between two input integer sequences as mentioned above.

Test your code with three different inputs for the two integer sequences such that one of the above three relationships are printed (i.e., the output is (i) for one pair of inputs, (ii) for another pair of inputs and (iii) for the third pair of inputs). Take screenshots of the outputs obtained for all the three pairs of input integer sequences.

You could choose the hash table size of your choice.

Q2 - 25 pts) Input two Linked Lists of integers (as Strings and then use String Tokenizer to extract the individual integers and construct the Linked Lists) and use a hash table to construct and print out the contents of an intersection Linked List that has the common elements of the two Linked Lists. The intersection Linked List should have the common elements appearing only once.

For example, if L1 = 4 --> 5 --> 2 --> 2 --> 3 --> 6

L2 = 6 --> 7 --> 2 --> 5 --> 2 --> 3

The intersection Linked List L12 = 2 --> 5 --> 3

Note that there are two instances of node '2' in both the Linked Lists L1 and L2, the intersection linked list should have only one instance of node '2'.

You are given the code discussed in class to find the union of two linked lists such that the union linked list has only unique integers even though the two linked lists may have duplicates. In this question too, the two linked lists may have duplicate integers, but the intersection linked list should have only unique integers.

Test your code with integer sequences of length 15, the maximum value for an integer being 20 and a hash table size of your choice. Capture the screenshot of your output.

Submission (through Canvas): Submit a single word document that contains the complete code for both the questions as well as the screenshots of the tests/outputs as indicated.