



CSCI 2270 – Data Structures

Midterm I Part II (Coding Problem)

Spring 2022: Friday, March 4, 5PM – 6PM

Instructions

1. This portion of the exam has one coding question worth 50 points. You have been provided with starter code.
2. Make sure to use the C++11 standard for compilation (`g++ -std=c++11`).
3. You need to write code for the function asked in the question. You can write helper functions if desired.
4. Read the specification given in the questions. Your code should be well-written in terms of commenting and indentation.
5. Describe your logic clearly in coding comments. In case your implementation is not fully correct, your comments may be helpful in getting you some partial credit.
6. You are required to upload a single file containing your solution to the specified Canvas submission link. Do not hand in any other files or upload a zipped file. You are responsible for ensuring that your file is attached properly. Technical excuses will not be accepted. Resubmits/late submissions will not be accepted.
7. In addition to being tested on several test-cases, the grading staff will also visually inspect your code for correctness, partial correctness, and memory leaks.
8. You are not allowed to distribute/share this exam in any way, shape, or form after the exam ends.
9. All students are expected to comply with the CU Boulder Honor Code Pledge: “On my honor, as a University of Colorado Boulder student, I have neither given nor received unauthorized assistance.”



CSCI 2270 – Data Structures

Midterm I Part II (Coding Problem)

Spring 2022: Friday, March 4, 5PM – 6PM

Question 1 (50% of overall exam grade)

You are provided with a linked list (LL) where each node contains an integer value. Your task is to complete the following two methods:

Starter Code and Submission:

You must complete the following functions:

```
void locateMaxMoveHead(int *&){...}  
void copyLLtoArr(int *&){...}
```

Starter code has been provided to you. You are only allowed to make changes to the `locateMaxMoveHead` and `copyLLtoArr` functions in `LinkedList.cpp`. However, you are not allowed to change the function parameters. Also, do not make any changes to `driver.cpp` or `LinkedList.hpp`.

Note that you are not guaranteed to receive full points just by passing the given test cases. Your algorithm also has to be correct and not introduce any memory leaks.

The code must compile with the following in a standard Linux or Mac terminal.

```
g++ -std=c++11 driver.cpp LinkedList.cpp
```

Specifications for `locateMaxMoveHead`:

This function moves the node with the largest key value to the front of the LL. Your solution should handle lists with 0, 1, and 2 elements and any other edge cases for that matter. The test cases do not contain duplicate keys/values so you do not need to check for them.

Note: You are not allowed to change the parameters of this function.

Specifications for `copyLLtoArr`:

Copies the LL node values to the array that is passed in to `copyLLtoArr` in the same order as the values exist within the LL. The array that is passed to this method has the



CSCI 2270 – Data Structures

Midterm I Part II (Coding Problem)

Spring 2022: Friday, March 4, 5PM – 6PM

correct length. This function **must** be called by `locateMaxMoveHead`.

Note: You are not allowed to change the parameters of this function.

Important Requirements:

Your solution will not be accepted if any of the following requirements are violated:

- **You are not allowed to delete nodes and re-create them. You must use the original nodes.**
- **You are not allowed to change node values. You must re-direct/re-assign the pointers as needed.**
- **You are not allowed to use any other arrays, a vector, or any of the string library functions.**
- **You are not allowed to use the Internet, use your phone, or communicate with anyone.**
- **You are not allowed to hard-code any values or manipulate the test cases so that they pass. Such occurrences will be considered as cheating.**

Example A

Number of nodes = 6

Input

2 → 5 → 13 → 9 → 7 → 8 → NULL

Output

For the Linked List, 13 → 2 → 5 → 9 → 7 → 8 → NULL

For the array, `arr[0] = 13`, `arr[1] = 2`, `arr[2] = 5`, `arr[3] = 9`, `arr[4] = 7`, `arr[5] = 8`

13 is the largest value in the LL, so it's moved to the front (head). All of the other elements are shifted over by one.



CSCI 2270 – Data Structures

Midterm I Part II (Coding Problem)

Spring 2022: Friday, March 4, 5PM – 6PM

Example B

Number of nodes = 9

Input

1 → 2 → 3 → 4 → 5 → 6 → 7 → 8 → 9 → NULL

Output

For the Linked List, 9 → 1 → 2 → 3 → 4 → 5 → 6 → 7 → 8 → NULL

For the array, arr[0] = 9, arr[1] = 1, arr[2] = 2, arr[3] = 3, arr[4] = 4, arr[5] = 5, arr[6] = 6, arr[7] = 7, arr[8] = 8

9 is the largest value in the LL, so it's moved to the front (head). All of the other elements are shifted over by one.