

CSCI 2270 Practice Midterm 1

Below are sample coding questions. These will give you an idea of what the difficulty level for the coding portion of the exam will be. You will need to use VS Code (C++11) to write your code just as you have been doing for your homework. You will not use Coderunner on the exam. Once you're finished, you will submit your C++ solution to the Midterm Coding question submission link within Canvas.

Problem 1

Task:

Write a program that creates a linked list of integers, and a function that returns `true` if the number of nodes in the list is even, and `false` if it is odd.

Requirements:

1. Implement a `lengthIsEven` function:

```
bool lengthIsEven(Node *head); // example declaration
```

This function should return `true` if the number of nodes in the linked list is even, and `false` if it is odd.

Examples:

* If the list is `3 -> 9 -> 2 -> 4 -> NULL`, calling

`lengthIsEven(head)` should return `true`.

* If the list is `3 -> 9 -> 2 -> NULL`, calling `lengthIsEven(head)` should return `false`.

* If the list is `NULL`, calling `lengthIsEven(head)` should return `true`.

2. Write a main function that creates a linked list of integers, then calls the function defined in part (1) to determine if the length of the list is even or odd. Call it on multiple lists to show that it works in all cases, and print out the results.
3. **Test your function** to make sure that it works in every case, no matter how many nodes are in the linked list.

Problem 2

Task:

Write a program that creates a linked list of integers, and a function that deletes every node with a negative value in that list.

Requirements:

1. Implement a `deleteNegatives` function:

```
Node *deleteNegatives(Node *head); // example declaration
```

This function should delete every negative node in the linked list and return a pointer to the new head of the list.

Examples:

- * If the list is `-3 -> 9 -> 2 -> -4 -> NULL`, calling `deleteNegatives(head)` should change the list to be `9 -> 2 -> NULL` and return a pointer to the new head.
- * If the list is `7 -> 6 -> 5 -> NULL`, calling `deleteNegatives(head)` should leave the list the same and return a pointer to the old head.

2. Write a main function that creates a linked list of integers, then calls the function defined in part (1) to delete all the negative

values. Call it on multiple lists to show that it works in all cases, and print out the results.

3. **Test your function** to make sure that it works in every case, no matter how many nodes or negative values are in the linked list.