

# ELEC 278: Fundamentals of Information Structures

## Lab 2: Working with Linked Lists

Fall 2023-Instructors: Ni & Mertin

September 19, 2023

### 1 Objectives

The objectives of this lab are to build your intuition around manipulating data structures by creating algorithms to make simple changes to linked lists.

### 2 Instructions

Download the file `lab2.zip` from OnQ and unzip it. Open the `lab2` folder that you extracted in either CLion or VS Code (note: you need to make sure you open the correct folder, i.e., the one that directly contains the code files). Then, complete the following tasks.

**Note that, for all tasks, to receive full marks, your code must be able to correctly handle any list of any length.**

#### 2.1 Task 1: Construct and Traverse a Linked List

The starter code includes the data structure definitions for a linked list (the same code as for problem 2 of the week 2 problem set). Comments indicate where you need to add additional code. For the first task, you need to:

1. Add code to `main()` to construct a linked list containing the items 1, 2, 3, 3, 4, 5, 6.
2. Implement the function `print_list()` to print the contents of the list in this format (including a newline at the end):  
1, 2, 3, 3, 4, 5, 6

#### 2.2 Task 2: Swap Adjacent Items in a Linked List

Implement the function `swap_adjacent()`, which swaps adjacent items in the linked list, ignoring the last item if there are an odd number of items. After doing this, the list should be:

2, 1, 3, 3, 5, 4, 6

#### 2.3 Task 3: Duplicate Full Linked List

Implement the function `double_list()`, which duplicates all items in the list, end to end. Starting from the result of task 2, the list should be:

2, 1, 3, 3, 5, 4, 6, 2, 1, 3, 3, 5, 4, 6

## 2.4 Task 4: Remove Adjacent Duplicate Items in a Linked List

Implement the function `remove_adjacent_duplicates()`, which removes adjacent items which are duplicates. Starting from the result of task 3, the list should be:

2, 1, 3, 5, 4, 6, 2, 1, 3, 5, 4, 6

Note: the code should be written such that, if there were several identical items in a row, they would all be replaced by a single copy (i.e., 1, 1, 1 would become just 1).

## 3 Marking Criteria

**After completing all tasks, call over a graduate TA to mark the lab.** Lab 2 has 4 marks in total:

- Does your code correctly generate the desired list and print it out? (1 mark)
- Does your code correctly switch adjacent nodes in the list? (1 mark)
- Does your code correctly duplicate the list end to end? (1 mark)
- Does your code correctly remove adjacent duplicate items in the list? (1 mark)