# Part 1 Linked List with O(1) Tail Access (30 marks)

You are provided with the trimmed skeleton code of Assignment 1. Your job is to implement the following three features. Note that, You will only get full marks if you can do the tasks in O(1) time.

## Task 1 Insert at Tail (15 marks)

Implement the function `insertTail(x)` that will insert an item `x` at the tail (last item) of the list in O(1) time. You will get full marks only if other functions such as `headItem()` and `removeHead()` still works.

## Task 2 Get the Middle Item (15 marks)

Implement the function `midItem()` to return the "middle" item in the linked list, that can be called at any time with O(1) running time. If there are n items in the list, the middle item is the $i^{th}$ item with

$$i = \text{floor}(n/2).$$

Your function should work perfectly with all the given functions in the given List class.

You can assume the list is not empty in Task 1 and Task 3. Here is an example:

```
List l;

for (int i = 0; i < 10; i++)
{
        l.insertTail(i * 2 + 1);
        cout << "List: ";
        l.print();
        cout << "Mid = " << l.midItem() << endl;
}
```

```
List: 1
Mid = 1
List: 1 3
Mid = 1
List: 1 3 5
Mid = 3
List: 1 3 5 7
Mid = 3
List: 1 3 5 7 9
Mid = 5
List: 1 3 5 7 9 11
Mid = 5
List: 1 3 5 7 9 11 13
Mid = 7
List: 1 3 5 7 9 11 13 15
Mid = 7
List: 1 3 5 7 9 11 13 15 17
Mid = 9
List: 1 3 5 7 9 11 13 15 17 19
Mid = 9

Remove Head
List: 3 5 7 9 11 13 15 17 19
Mid = 11
List: 5 7 9 11 13 15 17 19
Mid = 11
List: 7 9 11 13 15 17 19
Mid = 13
List: 9 11 13 15 17 19
Mid = 13
List: 11 13 15 17 19
Mid = 15
List: 13 15 17 19
Mid = 15
List: 15 17 19
Mid = 17
List: 17 19
Mid = 17
List: 19
Mid = 19
```

## Part 1 Submission

Copy and paste the entire `simpleLinkedList.cpp` file into Coursemology.