LABORATORY 7 : Singly Linked Lists / Doubly Linked List

OBJECTIVES

- to understand basic linked list operations
- to recognize the advantages of linked lists
- to be able to use and extend code written by other

BACKGROUND

Doubly linked list is a type of linked list that allows us to go in both directions – forward and reverse. The following is a doubly linked list of Strings representing airport codes. Please note that the linked list has sentinels.

Operations on a doubly linked list are carried out in the same manner as in a singly linked list. You just have to also pay attention to the 'prev' reference.

Object comparison in Python 3
In Python, "is" and "==" are both used for object comparison. The comparison operator "==" compares values of two objects. "is" returns when the two objects are same. Look at the execution below to see how "is" and "==" work.

```
>>> a = [1, 2, 3]
>>> b = a
>>> a
[1, 2, 3]
>>> b
[1, 2, 3]
>>> a == b
True
>>> a is b
True
>>> c = list(a)
>>> c
[1, 2, 3]
>>> a == c
True
>>> a is c
False
```

Python allows you to implement comparison for objects by operator overloading. Rich comparison methods assign a special method to each comparison operator (as shown in the table below). In order for comparing operators to work, their corresponding methods must be implemented in a class.

Operator	Method
==	__eq__
!=	__ne__
<	__lt__
<=	__le__
>	__gt__
>=	__ge__

The following web pages provide examples on how comparison methods are implemented in class.
https://portingguide.readthedocs.io/en/latest/comparisons.html
https://stackabuse.com/is-vs-python-object-comparison/

LABORATORY 7 : Pre-lab

1. Read book chapter on linked lists (Chapter 7) and doubly linked lists (section 7.3)
2. Study about linked lists from
   - https://runestone.academy/runestone/books/published/pythonds/BasicDS/ImplementinganUnorderedListLinkedLists.html
   - https://stackabuse.com/is-vs-python-object-comparison/

LABORATORY 7 : In-lab

1. Write `UnorderedList` class. You may try to implement it by yourself following https://runestone.academy/runestone/books/published/pythonds/BasicDS/ImplementinganUnorderedListLinkedLists.html or you can copy their code (try not to, though).
2. Add two methods to your `UnorderedList` class. They are `squish()` and `dble()`.
   - `squish()` takes this list and, wherever two or more consecutive items are equal, it removes duplicate nodes so that only one copy remains. Hence, no two consecutive items in this list are equal upon completion of the procedure.

     After `squish()` executes, the list maybe shorter than when `squish()` began. No extra items are added to make up for those removed.

     For example, if the input list is [0 0 0 0 1 1 0 0 0 3 3 3 1 1 0], the output list is [0 1 0 3 1 0].
   - `dble()` takes this list and doubles its length by adding a node after each node making the two nodes reference the same item.

     For example, if the input list is [3 7 4 2 2], the output list is [3 3 7 7 4 4 2 2 2 2].

     IMPORTANT: Do not try to create new copies of items. Just copy their references.

Note: You may need to write other `UnorderedList` methods (or data element) that are necessary for `squish()` and `dble()` to work.
For example, `insertAfter()` : insert a new node after a node pointing by the cursor, `deleteAfter()` : delete a node after a node pointing by the cursor, or `deleteAt()` : delete a node pointing by the cursor etc.

3. Write a test plan to test `squish()` and `dble()`. Implement your test plan in `ListTester.py` or prepare a test script for interactive testing.
4. Keep your linked list classes. You may get to use them again in the next labs.

LABORATORY 7 : Post-lab

Recording and Retrieving Golf Scores

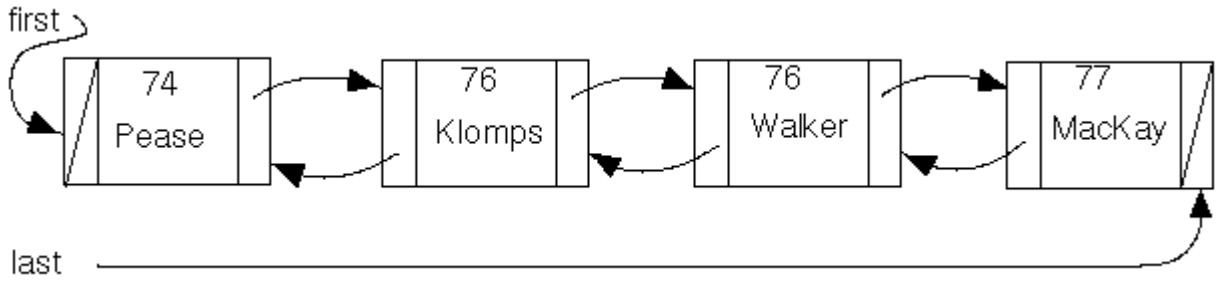
(ref: *http://www.cs.grinnell.edu/~walker/courses/161.sp12/modules/lists/reading-lists-double.shtml)*

In recording scores for a golf tournament, we enter the name and score of the player as the player finishes. This information is to be retrieved in each of the following ways:

- Scores and names can be printed in order by ascending or by descending scores.
- Given the name of a player, other players with the same score can be printed.

This problem requires the frequent updating of names and scores and the ordering of the information. Such requirements naturally suggest a linked list structure. However, here a simple linked list is difficult to use efficiently because all the pointers move in one logical direction from the start to the end. This structure makes it hard to move backward from a given name.

To resolve this difficulty, we consider a new type of structure, called a doubly linked list, in which each item contains a pointer to the previous item as well as to the next one. The following figure shows a doubly linked list for this problem.



Write a doubly linked list class for the above golf score scenarios. You should be able to

- add/delete/update score of a golf player
- print scores (in a way described above)
- retrieve scores (in a way described above)

Submission:

In-lab : Wednesday Oct 7 by 12pm

Post-lab : Wednesday Oct 14 at 9am

You are to demonstrate your test plan and program. Prepare to answer some questions individually.