

## LAB05: Addition using doubly linked list

CSE2050 Fall 2017

Instructor: Jeffrey Meunier & Wei Wei

TA: Jenny Blessing, Param Bidja, Yamuna Rajan & Zigeng Wang

### 1. Introduction

In this lab, we use Doubly Linked Lists to store positive integers and perform addition operations.

### 2. Objectives

The purpose of this assignment is to help you

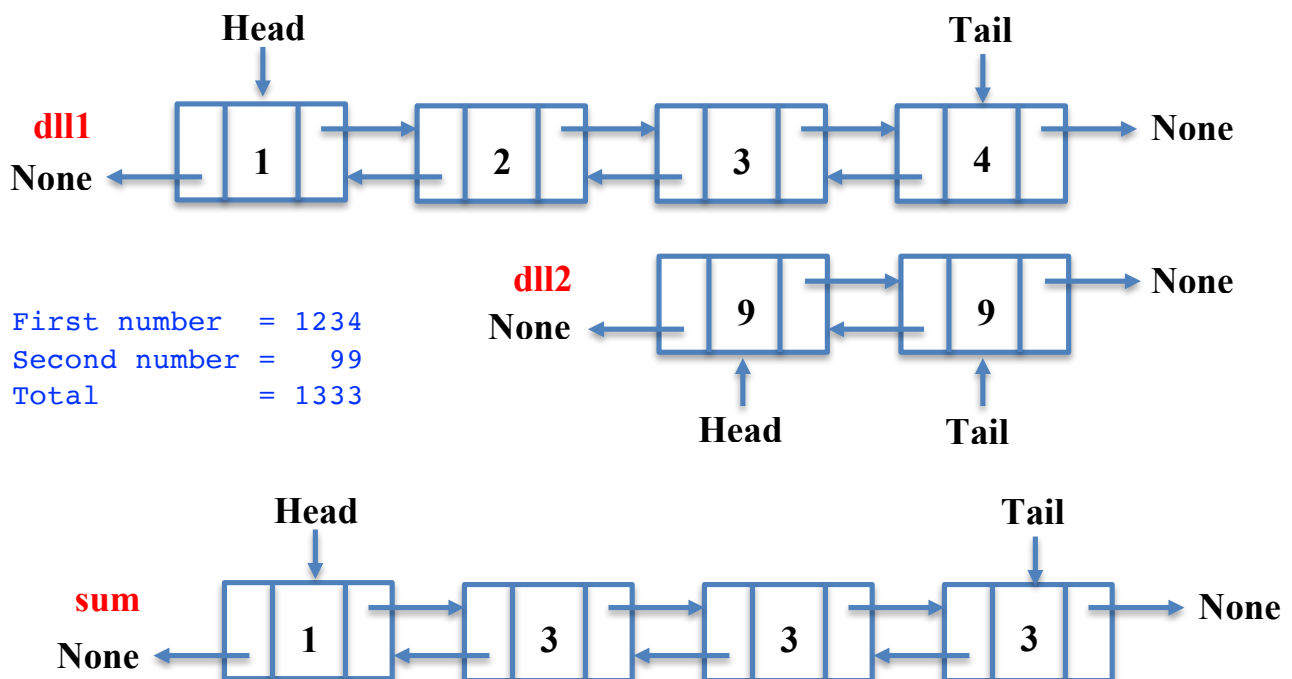
- ☐ Enhance your understanding of doubly linked lists
- ☐ Practice how to perform operations on doubly linked lists

**Note:** Before you start, if you are not familiar with DoublyLinkedLists you are recommended to review the sample codes we covered in lecture first.

### 3. Background

The purpose of this assignment is to input two positive integers, store each number in a doubly linked list (where each digit is data in a node), and perform addition on them by implementing various operations of doubly linked list.

For example, the first two doubly linked lists below represent 1234, and 99 respectively. The addition of these two numbers: 1333, is represented as the third doubly linked list.



## 4. Assignment

In the skeleton zip file, you can find a skeleton code. All you need to do is to complete the skeleton code based on the following instructions and submit the to Mimir.

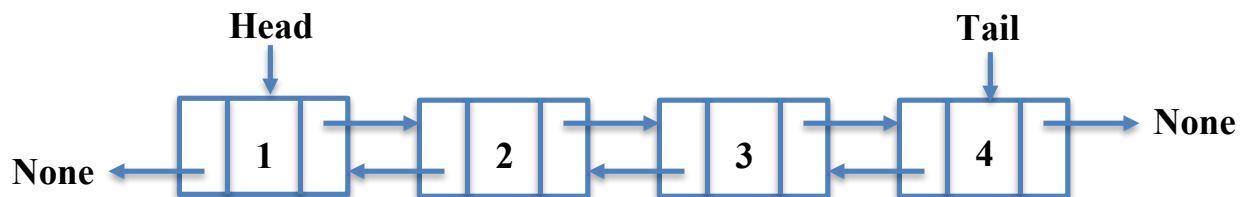
### 4.1 class DoublyLinkedList

This class has two methods –

- `tolinkednumber(self, string)` : In this method, each digit in the input string is added as data in a node in the doubly linked list. A for loop is used to iterate over the entire string. Each character in the string is added to the last of the doubly linked list.
- `__str__(self)` : This method returns the string representation of the object. Each node is converted to string by traversing through the entire doubly linked list.

An example:

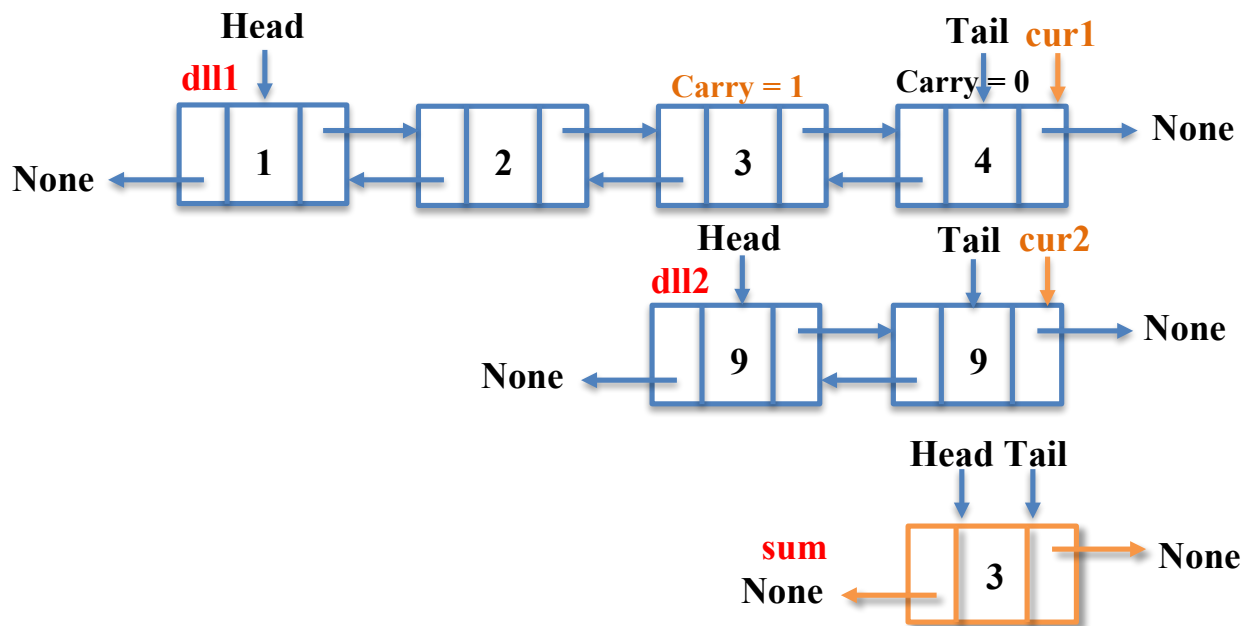
```
n = DoublyLinkedList()  
n.tolinkednumber('1234')
```



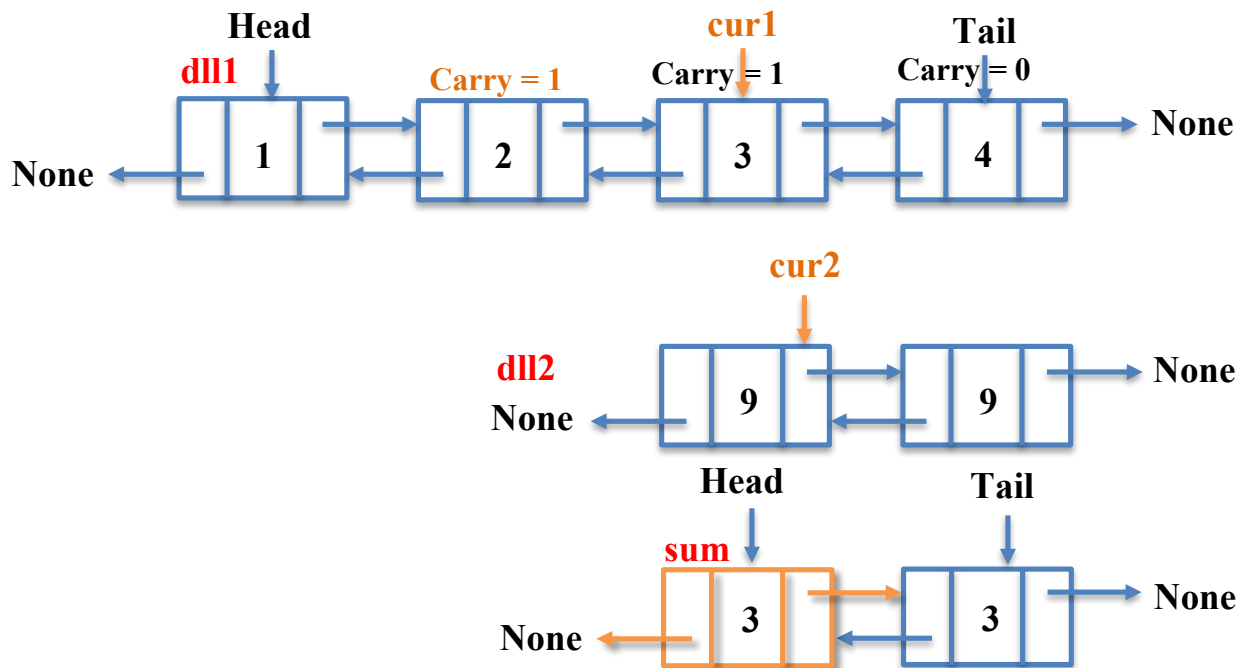
### 4.2 sumlinkednumbers(dll1, dll2)

This function performs the addition of the two doubly linked lists and stores the sum in a third doubly linked list. This function returns the sum.

1. The first step in addition is to start from the unit's place. Therefore, in our case, we must start from the tail of the two doubly linked lists. To do this we use `cur1` and `cur2` which point to the tail of the first and second doubly linked list respectively. In each iteration, they are updated to point to the previous node. We must also ensure that we account for the carry that will be generated during this process.

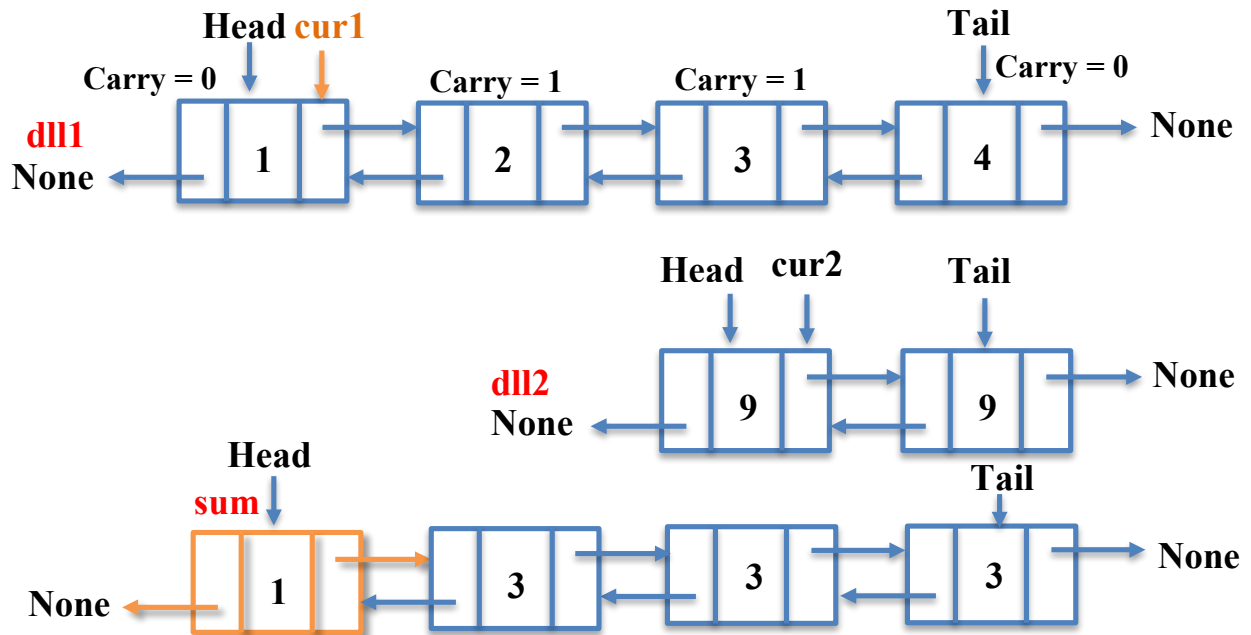


- The next step is to perform the actual addition. While cur1 and cur2 still point to a node, we add the digits in those nodes along with the carry. We must next calculate the new value of carry. Finally, the values of cur1 and cur2 are updated to the previous node.



- There are two more cases that we have to account for. Either the first number could have more digits than the second (in which case, cur2 would have reached the end of the list while cur1 would still point to a valid node) or vice versa. In this case, we simply

add the carry and valid node to generate the sum. We then calculate the new value of carry and update cur1 (or cur2) to point to the previous node.



You output should look like this:

```
The first number:
1234
The second number:
99
1234 + 99 = 1333
```

## 5. Submit your work to Mimir

Submit doublylinkednumber.py to Mimir after you complete your code. The Mimir will automatically grade your submission based on different unit tests. You can submit your code to Mimir up to **30 times** to refresh your existing score before the submission deadline.

## 6. Due date

This lab assignment is worth 2 points in the final grade. It will be due by 11:59pm on Wed. Oct 18<sup>th</sup> 2017. A penalty of 10% per day will be deducted from your grade, starting at 12:00am.

## 7. Getting help

Start your project early, because you will probably not be able to get timely help in the last few hours before the assignment is due.

- ☐ Go to the office hours of instructors and TAs.
  - Prof. Wei Wei: Mon. 2:30 - 3:15pm, Wed. 2:30 - 3:15pm, Thurs. 2:30 - 3:15pm, Fri. 2:30 - 3:15pm @ITE258
  - Jenny Blessing: Fri. 12pm - 2pm @ITE140

- Param Bidja: Tues. 2pm - 3pm @ITE140
  - Yamuna Rajan: Tues. 11am - 12pm, Wed. 9:30am – 10:30am @ITE140
  - Zigeng Wang: Mon. 3pm - 4pm @ITE140
- Post your questions on Piazza. TAs and many of your classmates may answer your questions.
- Search the answer of your unknown questions on the Internet. Many questions asked by you might have been asked and answered many times online already.