Exercise: Sum Two Linked Lists

Challenge yourself with the following exercise in which you'll sum up two linked lists!

Problem

In this exercise, you are required to sum two linked lists and return the sum embedded in another linked list.

The first number that we append to the linked list represents the unit place and will be the least significant digit of a number. The next numbers appended to the linked list will subsequently represent the *tenth, hundredth, thousandth,* and so on places.
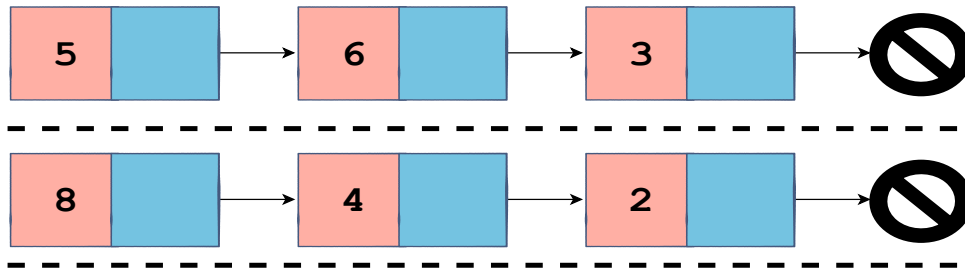
For example,

```
llist1 = LinkedList()
llist1.append(5)
llist1.append(6)
llist1.append(3)
```

in the code above, `llist1` represents the number `365`.

Note that you will not be tested on numbers whose sum requires an additional digit.

Below is an illustration to make the task clearer to you:

As you can see from the illustration above, we have two linked lists from which we calculate the sum, and the third linked list represents that sum.

Coding Time!

In the code below, the `sum_two_lists` is a class method of the `LinkedList` class. You cannot see the rest of the code as it is hidden. As `sum_two_lists` is a class method, please make sure that you don't change the indentation of the code provided to you. You are required to write your solution under the method prototype.

The two linked lists that you have to sum are `llist` and the linked list on which the method `sum_two_lists` is called. You will return the third linked list from the method where the data values in that linked list will represent the sum.

Keep in mind that the linked lists can be of different sizes.

Good luck!

```
def sum_two_lists(self, llist):
    pass
```