Cse 220 Lab Mid: Section 26
CO-1
Read The Question Carefully

Question 1 - Half Reversal, Half Sum :

You are given the ***head*** node of a linked list. The number of nodes/elements in the linked list will be even.

Your task is to:

- reverse the first half of the linked list
- calculate the sum of the elements in the last half of the linked list and add it at the end of the reversed linked list

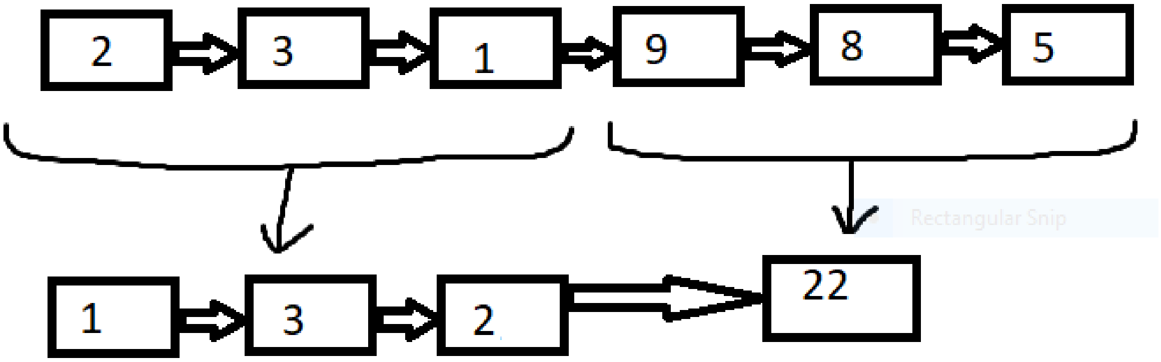
Hint: Use an out of place approach

Example:

Sample Input	Sample Output
2 -> 3 -> 1 -> 9 -> 8 -> 5	1 -> 3 -> 2 -> 22

Explanation: First half elements of the linked list are 2, 3, 1 respectively. Their reverse is 1, 3, 2. Second half elements are 9, 8, 5 whose total summation is 22.

Final result is reversed linked list + summation = 1, 3, 2, 22



Question 2 - Matrix Addition:

You are given two ***m x n*** matrices in the form of a 2D array. Your task is to perform a matrix addition operation. Both matrices have the same number of rows and columns, but the number of rows doesn't have to be the same as the number of columns.

```python
def add_matrices(matrix_1, matrix_2):
      #write your code here
```

<table>
<tr><td>Sample Input 1</td><td>Sample Output 1</td></tr>
<tr><td>
<table>
<tr><td>7</td><td>4</td></tr>
<tr><td>5</td><td>1</td></tr>
<tr><td>3</td><td>3</td></tr>
</table>
<table>
<tr><td>5</td><td>4</td></tr>
<tr><td>8</td><td>2</td></tr>
<tr><td>1</td><td>0</td></tr>
</table>
</td><td>
<table>
<tr><td>12</td><td>8</td></tr>
<tr><td>13</td><td>3</td></tr>
<tr><td>4</td><td>3</td></tr>
</table>
</td></tr>
<tr><td>Sample Input 2</td><td>Sample Output 2</td></tr>
<tr><td>
<table>
<tr><td>3</td><td>3</td><td>2</td><td>6</td></tr>
</table>
<table>
<tr><td>4</td><td>5</td><td>2</td><td>3</td></tr>
</table>
</td><td>
<table>
<tr><td>7</td><td>8</td><td>4</td><td>9</td></tr>
</table>
</td></tr>
</table>

Bonus Task - Remove Element:

Consider the following Node class of a Singly Linked List

```
class Node:

def __init__(self, data, next):
      self.data = data
      self.next = next
```

```
def remove_element(target_node):
      # write your code here
```

Using the above node class suppose a Linked List has been created with elements [2, 3, 6, 1, 9, 8, 5]

You were given the reference of one of the nodes from the linked list, you need to remove that element from the linked list.

Example: the method ***remove_element()*** is called and the reference to node containing 6 is being passed to the parameter: ***target_node***. Your task is to remove the element 6 from the linked list

Constraints: You cannot use loop or recursion to solve this problem. You do not have access to the head node, but only to the ***target_node***

Hint: It is ensured that ***target_node*** will not contain the reference to the last node of the linked list

For your understanding, here is a driver code provided

```
node = Node(5, None)
node = Node(8, node)
node = Node(9, node)
node = Node(1, node)
randomly_picked_node = node = Node(6, node)
node = Node(3, node)
node = Node(2, node)

def traverse_list(head_node: Node) -> Node:

    empty_string = ''

    result: str = empty_string

    node_cursor: Node = head_node

    while node_cursor is not None:

        if result is not empty_string:
            result += ' -> '

        result += str(node_cursor._node_data_value)

        node_cursor = node_cursor._reference_to_next_node

    return result


traverse_list(node) # expected output: 2 -> 3 -> 6 -> 1 -> 9 -> 8 -> 5
remove_element(randomly_picked_node)
traverse_list(node) # expected output: 2 -> 3 -> 1 -> 9 -> 8 -> 5
```