# Problem Challenge 2

We'll cover the following



## Rearrange a LinkedList (medium)

Given the head of a Singly LinkedList, write a method to modify the LinkedList such that the **nodes from the second half of the LinkedList are inserted alternately to the nodes from the first half in reverse order**. So if the LinkedList has nodes 1 -> 2 -> 3 -> 4 -> 5 -> 6 -> null, your method should return 1 -> 6 -> 2 -> 5 -> 3 -> 4 -> null.

Your algorithm should not use any extra space and the input LinkedList should be modified in-place.

**Example 1:**

```
Input: 2 -> 4 -> 6 -> 8 -> 10 -> 12 -> null
Output: 2 -> 12 -> 4 -> 10 -> 6 -> 8 -> null
```

**Example 2:**

```
Input: 2 -> 4 -> 6 -> 8 -> 10 -> null
Output: 2 -> 10 -> 4 -> 8 -> 6 -> null
```

## Try it yourself

Try solving this question here:

Java | Python3 | JS | C++

```python
from __future__ import print_function


class Node:
  def __init__(self, value, next=None):
    self.value = value
    self.next = next

  def print_list(self):
    temp = self
    while temp is not None:
      print(str(temp.value) + " ", end='')
      temp = temp.next
    print()


def reorder(head):
  # TODO: Write your code here
  return


def main():
  head = Node(2)
  head.next = Node(4)
  head.next.next = Node(6)
  head.next.next.next = Node(8)
  head.next.next.next.next = Node(10)
  head.next.next.next.next.next = Node(12)
  reorder(head)
  head.print_list()


main()

```


Back
Solution Review: Problem Challenge 1

Next
Solution Review: Problem Challenge 2

Completed