# Problem 2674: Split a Circular Linked List

## Problem Information

**Difficulty:** Medium
**Acceptance Rate:** 77.44%
**Paid Only:** Yes
**Tags:** Linked List, Two Pointers

## Problem Description

Given a **circular linked list** `list` of positive integers, your task is to split it into 2 **circular linked lists** so that the first one contains the **first half** of the nodes in `list` (exactly `ceil(list.length / 2)` nodes) in the same order they appeared in `list`, and the second one contains **the rest** of the nodes in `list` in the same order they appeared in `list`.

Return _an array answer of length 2 in which the first element is a**circular linked list** representing the **first half** and the second element is a **circular linked list** representing the **second half**._

A **circular linked list** is a normal linked list with the only difference being that the last node's next node, is the first node.

**Example 1:**

**Input:** nums = [1,5,7] **Output:** [[1,5],[7]] **Explanation:** The initial list has 3 nodes so the first half would be the first 2 elements since ceil(3 / 2) = 2 and the rest which is 1 node is in the second half.

**Example 2:**

**Input:** nums = [2,6,1,5] **Output:** [[2,6],[1,5]] **Explanation:** The initial list has 4 nodes so the first half would be the first 2 elements since ceil(4 / 2) = 2 and the rest which is 2 nodes are in the second half.

**Constraints:**

* The number of nodes in `list` is in the range `[2, 105]` * `0 <= Node.val <= 109` * `LastNode.next = FirstNode` where `LastNode` is the last node of the list and `FirstNode` is the first one

## Code Snippets

### C++:

```
/**
* Definition for singly-linked list.
* struct ListNode {
* int val;
* ListNode *next;
* ListNode() : val(0), next(nullptr) {}
* ListNode(int x) : val(x), next(nullptr) {}
* ListNode(int x, ListNode *next) : val(x), next(next) {}
* };
*/
class Solution {
public:
vector<ListNode*> splitCircularLinkedList(ListNode* list) {

}
};
```

### Java:

```
/**
* Definition for singly-linked list.
* public class ListNode {
* int val;
* ListNode next;
* ListNode() {}
* ListNode(int val) { this.val = val; }
* ListNode(int val, ListNode next) { this.val = val; this.next = next; }
* }
*/
class Solution {
public ListNode[] splitCircularLinkedList(ListNode list) {

}
}
```

### Python3:

```
# Definition for singly-linked list.
# class ListNode:
# def __init__(self, val=0, next=None):
# self.val = val
# self.next = next
class Solution:
def splitCircularLinkedList(self, list: Optional[ListNode]) ->
List[Optional[ListNode]]:
```