Problem Challenge 1

Palindrome LinkedList (medium)

Given the head of a **Singly LinkedList**, write a method to check if the **LinkedList is a palindrome** or not.

Your algorithm should use **constant space** and the input LinkedList should be in the original form once the algorithm is finished. The algorithm should have $O(N)$ time complexity where 'N' is the number of nodes in the LinkedList.

Example 1:

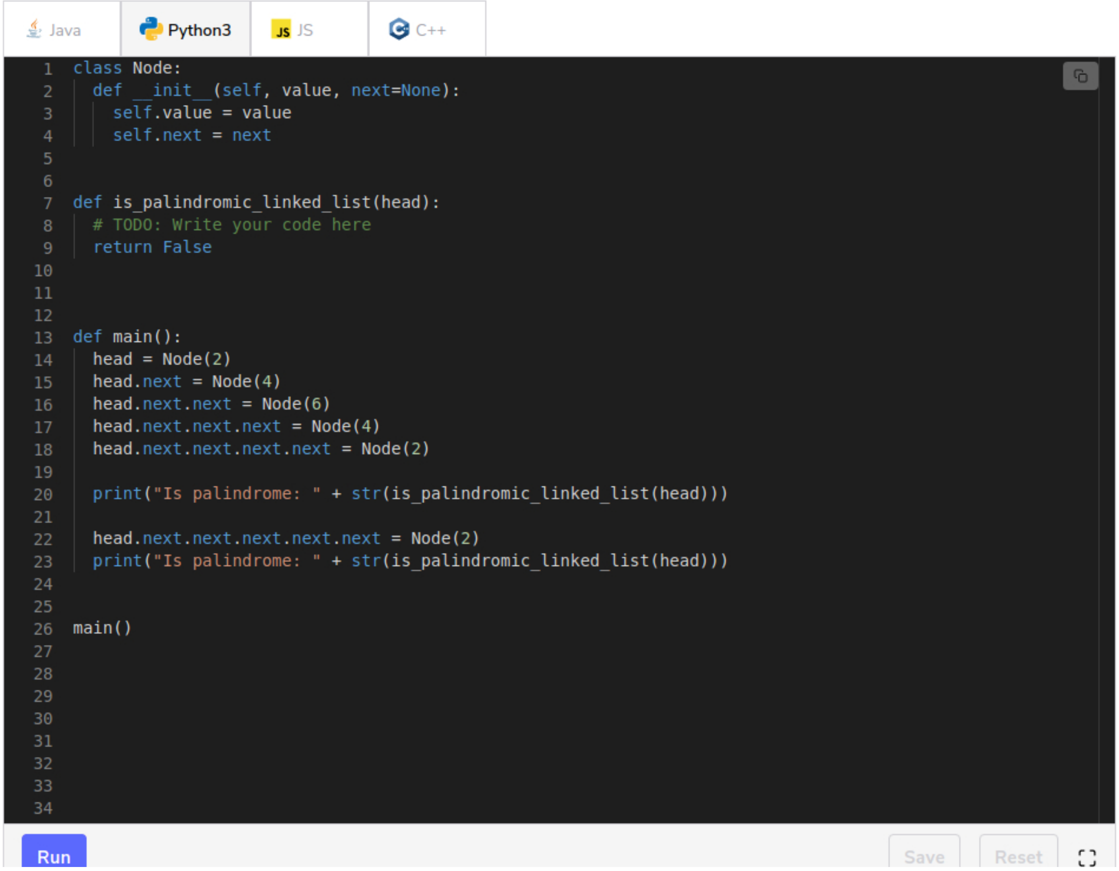
```
Input: 2 -> 4 -> 6 -> 4 -> 2 -> null
Output: true
```

Example 2:

```
Input: 2 -> 4 -> 6 -> 4 -> 2 -> 2 -> null
Output: false
```

Try it yourself

Try solving this question here:



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


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



def main():
  head = Node(2)
  head.next = Node(4)
  head.next.next = Node(6)
  head.next.next.next = Node(4)
  head.next.next.next.next = Node(2)

  print("Is palindrome: " + str(is_palindromic_linked_list(head)))

  head.next.next.next.next.next = Node(2)
  print("Is palindrome: " + str(is_palindromic_linked_list(head)))


main()
```

← Back

Next →

Middle of the LinkedList (easy)

Solution Review: Problem Challenge 1

☑ Completed

Report an Issue

Ask a Question