

Problem 5: Given the head of a [linked list](#), return *the node where the cycle begins*.
If there is no cycle, return null.

```
class ListNode:
    def __init__(self, val=0, next=None):
        self.val = val
        self.next = next

def detectCycle(head):
    if not head or not head.next:
        return None

    slow = head
    fast = head
    while fast and fast.next:
        slow = slow.next
        fast = fast.next.next
        if slow == fast:
            break
    else:
        return None

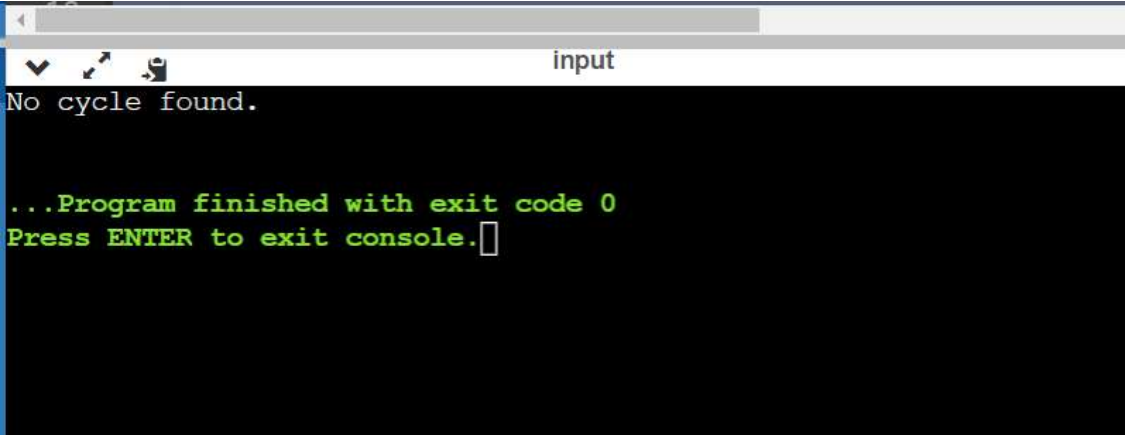
    fast = head
    while slow != fast:
        slow = slow.next
        fast = fast.next

    return slow

def createLinkedList(values):
    head = None
    current = None
    for val in values:
```

```
node = ListNode(val)
if not head:
    head = node
    current = head
else:
    current.next = node
    current = node
return head
```

```
values = [1, 2, 3, 4, 3, 6, 10]
head = createLinkedList(values)
result = detectCycle(head)
if result:
    print("tail connects to node index", values.index(result.val))
else:
    print("No cycle found.")
```

A screenshot of a terminal window titled "input". The terminal has a black background with white text. It displays the output "No cycle found." followed by a green message: "...Program finished with exit code 0" and "Press ENTER to exit console." with a cursor. The terminal window has a standard macOS-style title bar with a red, yellow, and green button on the left and a scroll bar on the right.

```
input
No cycle found.

...Program finished with exit code 0
Press ENTER to exit console.
```