

## 코딩 테스트 # 48

## 문제

단일 연결 리스트(singly linked list)가 주어지면 리스트의 중간 노드 값을 프린트 하시오.  
(제일 효율적인 방법으로)

Given a singly linked list, print the value of the node that is in the middle of the list.

풀이

이 문제를 가장 쉽게 푸는 방법은 리스트의 원소를 모두 체크하며 리스트의 길이를 안뒤, 새로운 루프로 길이의 절반까지만 전진한 후 값을 프린트 하면 됩니다.

하지만 그렇게 할 경우  $O(n) + O(n/2)$ 의 시간복잡도를 가지게 됩니다.

더 효율적인 방법은 포인터 두개를 만들어 리스트를 돌면 됩니다.

첫번째 포인터는 2칸씩 움직이고, 두번째 포인터를 1칸씩 움직입니다.

첫번째 포인터가 None/Null 의 값을 가질때는 리스트를 끝까지 돌았다는걸 말합니다.

이때 두번째 포인터는 리스트의 중간 노드를 가르키고 있습니다.

시간복잡도는  $O(n)$  입니다.

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

# node1 -> node2 -> node3 -> node4 -> node5 -> None
node5 = Node("e", None)
node4 = Node("d", node5)
node3 = Node("c", node4)
node2 = Node("b", node3)
node1 = Node("a", node2)

head = node1

fastPointer = head
slowPointer = head

while fastPointer.next != None and fastPointer.next.next != None:
    fastPointer = fastPointer.next.next
    slowPointer = slowPointer.next

print slowPointer.data
```

