MICP Week 5 Homework

#TEBOW IT

T-Talk:

How does the input look?

- input given by interviewer, usually a pointer to the head of the linked list.

What will the linked list contain?

- any Ascii character is allowed.

What if input is null?

- throw an exception.

What will the function return?

- true/false

How will the linked list be?

- singly linked list

E- Examples:

Sample Input	Equivalence class	Output
Null		exception
a	One element	True
a → b a	Two elements	True
a → b → a	Three elements	True
a → b	Two elements (fail)	False
a → b → c	Three elements (fail)	False
1 → 2 → 1	Numberical elements	True
$ → ! → $	Other characters	True
$ → a → 1	Mixed characters	False
A → b → a	Upper + lower case	False

B - Brute Force:

- Reverse and store the linked list
- Compare the elements of the original and reversed linked list
  - if all are same, return true
  - else, false
- This would be $O(n) + O(n) = O(2n) = O(n)$

O - optimize :

- Try to define one function which both reverses and compares instead of 2 separate function
- Try recursion

W - Walk Through :

① Check if head is NULL
- if it is, throw an exception

② The function will take 2 parameters - both head but one will be reversed

③ Recursively call the function to reverse the linked list and compare the values.

For example, for $a \to b \to c$

org: $a \to b \to c$

reverse: $c \to b \to a$

compare - $a \neq c$

$b \quad b$

$c \neq a$

return false

for $a \to b \to a$

org: $a \to b \to a$

reverse: $a \to b \to a$

compare: $a = a$

$b = b$

$a = a$

return true

I-Implement

```
bool isPalindrome(struct Node * org, struct Node *
                                    to_be_reversed)
{
    if (to_be_reversed == NULL)
            return true;
    bool check = isPalindrome(org, to_be_reversed->next);

    bool ans = (org->data == to_be_reversed->data);
    org = org->next;
    return ans;
}
```

T- Test