Doubly Linked List

You can assume your node class is given.

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

However, you may need to change the node class according to your problem.

Question - 01

Given a non-dummy headed doubly non-circular linked list, write a function that returns true if the given linked list is a palindrome, else false.

Example:
Input: 1 ⇔ 7 ⇔ 7 ⇔ 1 ⇔ None
Output: True

Input: 1 ⇔ 7 ⇔ 4 ⇔ 5 ⇔ None
Output: False

Question - 02

Given a non-dummy headed doubly non-circular linked list, reverse the list.

Example:
Input: 10 ⇔ 20 ⇔ 30 ⇔ 40 ⇔ 50 ⇔ None
Output: 50 ⇔ 40 ⇔ 30 ⇔ 20 ⇔ 10 ⇔ None

Question - 03

Given a dummy headed doubly circular linked list, find the largest node in the doubly linked list.

Example:
Input: dummy_head ⇔ 10 ⇔ 70 ⇔ 40 ⇔ 15 ⇔ dummy_head (consider it circular)
Output: 70

Question - 04

Given a dummy headed doubly circular linked list, rotate it left by k node (where k is a positive integer)

Question - 05

Given a dummy headed doubly circular linked list, rotate it right by k node (where k is a positive integer)