**Time: 25 minutes  
  
Question:**

**Rearrange a Doubly Linked List with Odd and Even Elements**

**Given a doubly linked list, rearrange it so that all odd elements appear before all even elements, while maintaining the relative order of the nodes.**

**The first node of the resultant modified doubly linked list should always be an odd number if there is at least one odd element. Otherwise, the order remains the same.**

**You can assume the Node class is already defined, with attributes elem, next, and prev.**

**Note: You can only use linked list data structure**

| **Sample Input** | **Sample Output** |
| --- | --- |
| **5 ⇄ 2 ⇄ 8 ⇄ 1 ⇄ 4 ⇄ 7** | **5 ⇄ 1 ⇄ 7 ⇄ 2 ⇄ 8 ⇄ 4** |
| **7 ⇄ 9 ⇄ 2 ⇄ 4** | **7 ⇄ 9 ⇄ 2 ⇄ 4** |
| **6 ⇄ 2 ⇄ 8 ⇄ 10 ⇄ 12** | **6 ⇄ 2 ⇄ 8 ⇄ 10 ⇄ 12** |

**Time: 25 minutes**

**Question:**

**Rearrange a Doubly Linked List with even and odd Elements**

**Given a doubly linked list, rearrange it so that all even elements appear before all odd elements, while maintaining the relative order of the nodes.**

**The first node of the modified resultant doubly linked list should always be an even number if there is at least one even element. Otherwise, the order remains the same.**

**You can assume the Node class is already defined, with attributes elem, next, and prev.**

**Note: You can only use linked list data structure**

| **Sample Input** | **Sample Output** |
| --- | --- |
| **5 ⇄ 2 ⇄ 8 ⇄ 1 ⇄ 4 ⇄ 7** | **2 ⇄ 8 ⇄ 4 ⇄ 5 ⇄ 1 ⇄ 7** |
| **7 ⇄ 9 ⇄ 2 ⇄ 4** | **2 ⇄ 4 ⇄ 7 ⇄ 9** |
| **5 ⇄ 9 ⇄ 13 ⇄ 11 ⇄ 15** | **5 ⇄ 9 ⇄ 13 ⇄ 11 ⇄ 15** |