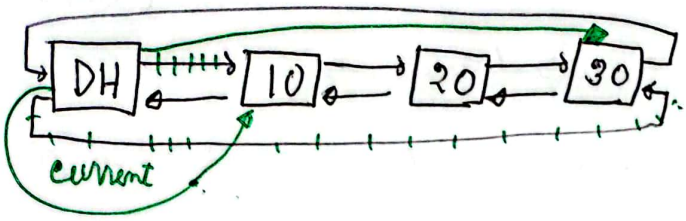
Doubly linked list reversal

IT-1



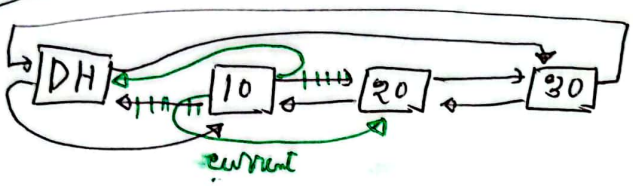
current = DH

swap [next = 10 (node)
prev = 30 (node)]

↳ After swap

[next = 30
prev = 20] after swap

IT-2

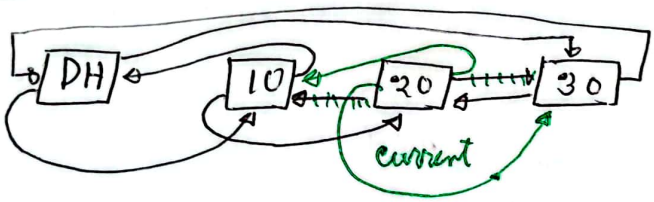


current = 10

next = 20
prev = DH } swap

next = DH
prev = 20 } After swap

IT-3

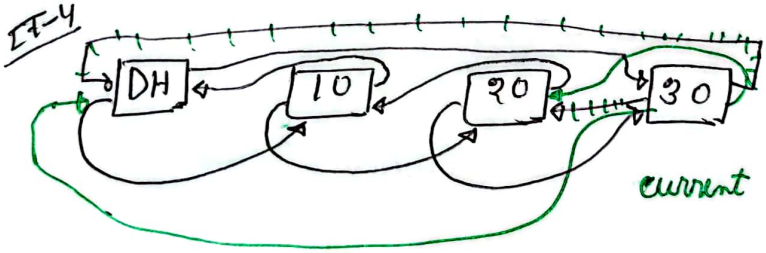


current = 20

next = 30
prev = 10 } swap

next = 10
prev = 30 } after swap

IT-4

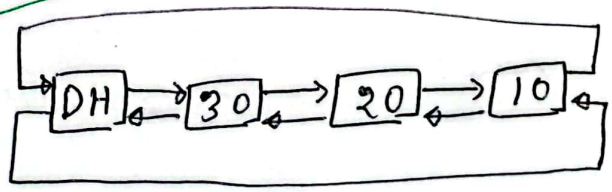


current = 30

next = DH } swap
prev = 20

next = 20
prev = DH } after swap

Redrawn image of Iteration-4



Swapping 2 values

```
temp = a
a = b
b = temp
```

Swapping next & prev value of a node (current_node)

```
temp = current_node.next
current_node.next = current_node.prev
current_node.prev = temp
```

(here) ಏನಾಗತ್ತೆ, a = current_node.next

b = current_node.prev

```
def reverseLL(head):
    if head.next == head:
        print("empty LL")
        return
    current_node = head
    while True:
        # swapping next & prev of current
        temp = current_node.next
        current_node.next = current_node.prev
        current_node.prev = temp
        current_node = current_node.prev
        # completed 1 full cycle.
        if current_node == head:
            break
```

Since next & previous have been swapped, in order to move forward i = i.prev instead of i = i.next