Day 12

Circular Singly linked list

Why circular?

In a singly linked list, for accessing any node of linked list, we start traversing from the first node. If we are at any node in the middle of the list, then it is not possible to access nodes that precede the given node.

Problem programme:

```
Class Node:

    def __init__(self,data):
        self.data = data
        self.next = None
```

```
Class CircularList :
    def __init__(self):
        self.last = None
    def addedtoempty(self,data):
        if (self.last != None)
            return self.last
        temp = Node(data)
        self.last = temp
        self.last.next = self.last
        return self.last
    def addbegin(self,data):
        if (self.last == None):
            return self.addedtoempty(data)
        temp = Node(data)
        temp.next = self.last.next
        self.last.next = temp
        return self.last
```

```
def addend (self, data):
    if (self.last == None):
        return self.addedtoempty (data)
    temp = Node(data)
    temp.next = self.last.next
    self.last.next = temp
    self.last = temp
    return self.last

def addafter(self, data,
                item):
    if (self.last == None):
        return None
    temp = Node(data)
    p = self.last.next
    while p:
        if (p.data == item):
            temp.next = p.next
            p.next = temp
            if (p == self.last):
                self.last = temp
                return self.last
```

```
        else:
            self.last

    p = p.next
    if (p == self.last.next):
        print (item, "not present in the list")
        break
```

```
def traverse(self):
    if (self.last == None):
        print ("list is empty")
        return

    temp = self.last.next
    while temp:
        print(temp.data, end = " ")
        temp = temp.next
        if temp == self.last.next:
            break
```

```
L = circularlist()
last = L.addtoempty(6)
last = L.addbegin(4)
last = L.addbegin(2)
last = L.addend(8)
last = L.addend(12)
last = L.addafter(10,8)
L.traverse()
```

Output :

2 4 6 8 10 12