EX.NO:11(E) Singly Linked List: Add Element at the Start

This Python program demonstrates the implementation of a **Singly Linked List** where a new element can be added at the **start** of the list.

Aim

To write a Python program that adds a **new element** at the **start** of a singly linked list. The program implements a `push_front` method that inserts an element at the front of the list, followed by a method to print the list.

Algorithm

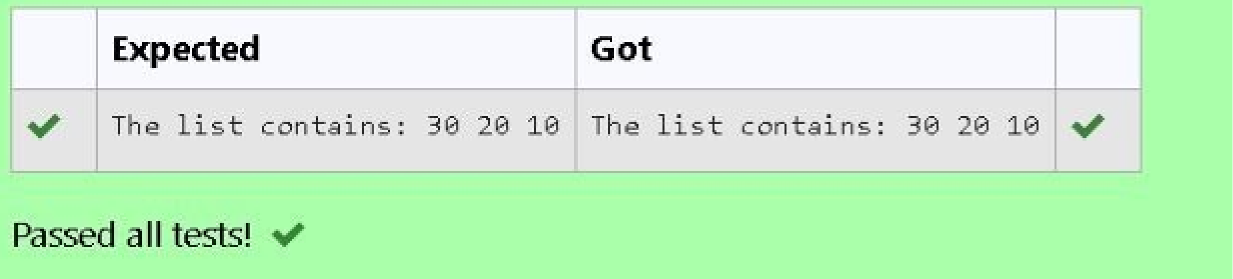
1. **Step 1:** Define a class `Node` with:
   - `data` to store the node's value.
   - `next` to store the reference to the next node.
2. **Step 2:** Define a class `LinkedList` with:
   - `head` to point to the first node.
3. **Step 3:** In the `LinkedList` class, define a method `push_front(newElement)`:
   - Create a new node with `newElement`.
   - Set the new node's next pointer to the current head node.
   - Set the head to the new node.

4. **Step 4:** Define a method `PrintList()` to display the list:
    - Print the elements of the list or display "The list is empty." if the list is empty.
5. **Step 5:** Instantiate a `LinkedList` object, `MyList`, and add elements at the start using the `push_front()` method.
6. **Step 6:** Call the `PrintList()` method to display the list.

Program

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

class LinkedList:
    def __init__(self):
        self.head = None

    def push_front(self, newElement):
        new_node = Node(newElement)
        new_node.next = self.head
        self.head = new_node

    def PrintList(self):
        temp = self.head
        if temp is not None:
            print("The list contains:", end=" ")
            while temp is not None:
                print(temp.data, end=" ")
                temp = temp.next
            print()
        else:
            print("The list is empty.")

# Create a linked list and add elements
MyList = LinkedList()
MyList.push_front(10)
MyList.push_front(20)
MyList.push_front(30)
MyList.PrintList()
```

Output



	Expected	Got	
✓	`The list contains: 30 20 10`	`The list contains: 30 20 10`	✓

Passed all tests! ✓

Result

Thus the program has been successfully executed