# Doubly LinkedList

1. Write a Python program to implement  the Doubly Linked List and it includes the below operations
    - insert an element in the beginning
    - insert an element in the end
    - insert an element after an element
    - delete an element in the beginning
    - delete an element in the end
    - delete an element after an element
    - display the linked list

**Program**

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

class DoublyLinkedList:
    def _init_(self):
        self.head = None

    def is_empty(self):
        return self.head is None

    def insert_first(self, data):
        new_node = Node(data)
        if self.is_empty():
            self.head = new_node
        else:
            new_node.next = self.head
            self.head.prev = new_node
            self.head = new_node

    def insert_last(self, data):
        new_node = Node(data)
        if self.is_empty():
            self.head = new_node
        else:
            current = self.head
            while current.next is not None:
                current = current.next
            current.next = new_node
            new_node.prev = current
    def insert_after(self, target_data, data):
        new_node = Node(data)
        current = self.head
        while current:
            if current.data == target_data:
                new_node.next = current.next
                new_node.prev = current
                if current.next:
                    current.next.prev = new_node
                current.next = new_node
                return
            current = current.next
        print(f"Element {target_data} not found in the list.")

    def delete_first(self):
        if self.is_empty():
            print("The list is empty.")
        else:
            if self.head.next:
                self.head = self.head.next
                self.head.prev = None
            else:
                self.head = None

    def delete_last(self):
        if self.is_empty():
            print("The list is empty.")
        else:
            current = self.head
            while current.next is not None:
                current = current.next
            if current.prev:
                current.prev.next = None
            else:
                self.head = None

    def delete_after(self, target_data):
        current = self.head
        while current:
            if current.data == target_data:
                if current.next:
                    to_delete = current.next
                    current.next = to_delete.next
                    if to_delete.next:
                        to_delete.next.prev = current
                    return
```

```python
                else:
                    print(f"No element exists after {target_data}.")
                    return
            current = current.next
        print(f"Element {target_data} not found in the list.")

    def display(self):
        if self.is_empty():
            print("The list is empty.")
        else:
            current = self.head
            while current:
                print(current.data, end=' ')
                current = current.next
            print()

dll = DoublyLinkedList()
dll.insert_first(10)
dll.insert_last(30)
dll.insert_first(5)
dll.insert_after(10, 20)
dll.display()

dll.delete_first()
dll.display()

dll.delete_last()
dll.display()

dll.delete_after(10)
dll.display()
```

**Output**

```
5 10 20 30
10 20 30
10 20
10
```