

```
1 class Node:
2     def __init__(self, data):
3         self.data = data
4         self.next = None
5
6
7 class LinkedList:
8     def __init__(self):
9         self.head = None
10
11     def is_empty(self):
12         return self.head is None
13
14     def append(self, data):
15         new_node = Node(data)
16         if self.is_empty():
17             self.head = new_node
18             return
19         last_node = self.head
20         while last_node.next:
21             last_node = last_node.next
22         last_node.next = new_node
23
24     def prepend(self, data):
25         new_node = Node(data)
26         new_node.next = self.head
27         self.head = new_node
```

Stack

```
1 class Stack:
2     def __init__(self):
3         self.items = []
4
5     def is_empty(self):
6         return len(self.items) == 0
7
8     def push(self, item):
9         return self.items.append(item)
10
11    def pop(self):
12        if self.is_empty():
13            raise IndexError("Underflow detected")
14        return self.items.pop()
15
16    def peek(self):
17        if self.is_empty():
18            raise IndexError("Underflow detected")
19        return self.items[-1]
20
21    def size(self):
22        return len(self.items)
23
24    def __str__(self):
25        return str(self.items)
```



Queue

```
1 class Queue:
2     def __init__(self):
3         self.items = []
4
5     def is_empty(self):
6         return len(self.items) == 0
7
8     def enqueue(self, item):
9         self.items.append(item)
10
11    def dequeue(self):
12        if self.is_empty():
13            raise IndexError("Underflow detected")
14        return self.items.pop(0)
15
16    def front(self):
17        if self.is_empty():
18            raise IndexError("Underflow detected")
19        return self.items[0]
20
21    def size(self):
22        return len(self.items)
23
24    def __str__(self):
25        return str(self.items)|
```

PROBLEMS OUTPUT DEBUG CONSOLE PORTS GITLENS COMMENTS TERMINAL

```
-----Stack Implementation-----
Stack after pushing: [1, 2, 3, 4, 5]
Stack after popping: [1, 2, 3, 4]
Top element: 4
Size of the stack: 4
Popped element: 4
Popped element: 3
Popped element: 2
Popped element: 1
Is the stack empty? True
----- Implementation Over-----
-----Queue Implementation-----
Queue after enqueueing: [1, 2, 3, 4, 5]
Dequeued element: 1
Queue after dequeuing: [2, 3, 4, 5]
Front element: 2
Size of the queue: 4
Dequeued element: 2
Dequeued element: 3
Dequeued element: 4
Dequeued element: 5
Is the queue empty? True
----- Implementation Over-----
-----Linked List Implementation-----
0 -> 1 -> 2 -> 3 -> None
0 -> 1 -> 3 -> None
1 -> 3 -> None
1 -> None
----- Implementation Over-----
-----Doubly Linked List Implementation-----
1 2 3
0 1 2 3
0 1 3
0 1
----- Implementation Over-----
```