

2 STACKS

```
TermProject1TWOSTACKFinals_Jacoblvlan.py x TeamProject1STACKANDQFinals_Jacoblvlan.py LinkedStack.py LinkedQueue.py
1 from LinkedStack import LinkedStack as _LinkedStack
  1 usage
2 class TwoStackDeq:
3     def __init__(self):
4
5         self.front_stack = LinkedStack()
6         self.rear_stack = LinkedStack()
7
8     1 usage (1 dynamic)
9     def is_empty(self):
10         return self.front_stack.is_empty() and self.rear_stack.is_empty()
11
12     def __len__(self):
13         return len(self.front_stack) + len(self.rear_stack)
14
15     1 usage
16     def add_first(self, e):
17         self.front_stack.push(e)
18
19     def add_last(self, e):
20         self.rear_stack.push(e)
21
22     4 usages
23     def _transfer(self, source, target):
24         while not source.is_empty():
25             target.push(source.pop())
26
27     1 usage
28     def first(self):
29         if self.front_stack.is_empty():
30             if self.rear_stack.is_empty():
31                 raise Exception("Deque is empty")
32             self._transfer(self.rear_stack, self.front_stack)
33             return self.front_stack.top()
34
35     1 usage
36     def last(self):
37         if self.rear_stack.is_empty():
38             if self.front_stack.is_empty():
39                 raise Exception("Deque is empty")
40             self._transfer(self.front_stack, self.rear_stack)
41             return self.rear_stack.top()
```

```
TermProject1TWOSTACKFinals_Jacoblvlan.py x TeamProject1STACKANDQFinals_Jacoblvlan.py LinkedStack.py LinkedQueue.py
24 def first(self):
  1 usage
25     if self.front_stack.is_empty():
26         if self.rear_stack.is_empty():
27             raise Exception("Deque is empty")
28         self._transfer(self.rear_stack, self.front_stack)
29     return self.front_stack.top()
30
31 1 usage
32 def last(self):
33     if self.rear_stack.is_empty():
34         if self.front_stack.is_empty():
35             raise Exception("Deque is empty")
36         self._transfer(self.front_stack, self.rear_stack)
37     return self.rear_stack.top()
38
39 2 usages
40 def remove_first(self):
41     if self.front_stack.is_empty():
42         if self.rear_stack.is_empty():
43             raise Exception("Deque is empty")
44         self._transfer(self.rear_stack, self.front_stack)
45     return self.front_stack.pop()
46
47 1 usage
48 def remove_last(self):
49     if self.rear_stack.is_empty():
50         if self.front_stack.is_empty():
51             raise Exception("Deque is empty")
52         self._transfer(self.front_stack, self.rear_stack)
53     return self.rear_stack.pop()
54
55 D = TwoStackDeq()
56
57 for i in range(1,6):
58     D.add_first(i)
59
60 print(D.first())
61 print(D.last())
62
63 print(D.remove_first())
```

```
Run TermProject1TWOSTACKFinals_JacobIvan x
"C:\Program Files\Python312\python.exe" Z:\DSAL601-IDB2\FINALS\TermProjectFinals_JacobIvan\TermProject1TWOSTACKFinals_JacobIvan.py
5
1
5
1
1
4
```

STACK AND QUEUES

```
TermProject1TWOSTACKFinals_JacobIvan.py TeamProject1STACKANDQFinals_JacobIvan.py x LinkedStack.py LinkedQueue.py
1 from LinkedStack import LinkedStack
2 from LinkedQueue import LinkedQueue
3 class StackandQueue:
4
5     def __init__(self):
6         self.front_queue = LinkedQueue()
7         self.back_stack = LinkedStack()
8
9     1 usage
10    def len(self):
11        return len(self.front_queue) + len(self.back_stack)
12
13    5 usages (1 dynamic)
14    def is_empty(self):
15        return self.len() == 0
16
17    1 usage
18    def add_first(self, e):
19        self.front_queue.enqueue(e)
20
21    def add_last(self, e):
22        self.back_stack.push(e)
23
24    2 usages
25    def delete_first(self):
26        if self.is_empty():
27            raise Exception('Deque is empty')
28        if self.front_queue.is_empty():
29            while not self.back_stack.is_empty():
30                self.front_queue.enqueue(self.back_stack.pop())
31        return self.front_queue.dequeue()
32
33    1 usage
34    def delete_last(self):
35        if self.is_empty():
36            raise Exception('Deque is empty')
37        if self.back_stack.is_empty():
38            while not self.front_queue.is_empty():
39                self.back_stack.push(self.front_queue.dequeue())
40        return self.back_stack.pop()
```

```
TermProject1TWOSTACKFinals_JacobIvan.py TeamProject1STACKANDQFinals_JacobIvan.py x LinkedStack.py LinkedQueue.py
26         self.front_queue.enqueue(self.back_stack.pop())
27     return self.front_queue.dequeue()
28
29     1 usage
30     def delete_last(self):
31         if self.is_empty():
32             raise Exception('Deque is empty')
33         if self.back_stack.is_empty():
34             while not self.front_queue.is_empty():
35                 self.back_stack.push(self.front_queue.dequeue())
36         return self.back_stack.pop()
37
38     1 usage
39     def first(self):
40         if self.is_empty():
41             raise Exception('Deque is empty')
42         if self.front_queue.is_empty():
43             while not self.back_stack.is_empty():
44                 self.front_queue.enqueue(self.back_stack.pop())
45         return self.front_queue.first()
46
47     1 usage
48     def last(self):
49         if self.is_empty():
50             raise Exception('Deque is empty')
51         if self.back_stack.is_empty():
52             while not self.front_queue.is_empty():
53                 self.back_stack.push(self.front_queue.dequeue())
54         return self.back_stack.top()
55
56 D = StackandQueue()
57
58 for i in range(1,6):
59     D.add_first(i)
60
61 print(D.first())
62 print(D.last())
63 print(D.delete_first())
64 print(D.delete_last())
65 print(D.delete_first())
66
```

```
Run TeamProject1STACKANDQFinals_JacobIvan x
C:\Program Files\Python312\python.exe Z:\DSAL601-IDB2\FINALS\TermProjectFinals_JacobIvan\TeamProject1STACKANDQFinals_JacobIvan.py
1
5
5
1
2
Process finished with exit code 0
```