

Name	Jhelo D. Palco	Date Submitted	Sep 10, 2024
Course & Year	BSCS-2	Instructor	Sir Jeremy Moses Ebreo

Suppose you have a **deque D** containing the numbers (1,2,3,4,5,6,7,8), in this order. Suppose further that you have an initially **empty queue Q**. Give a code fragment that uses only D and Q (and no other variables) and results in D storing the elements in the order (1,2,3,5,4,6,7,8).

Repeat the previous problem using the **deque D** and an initially **empty stack S**.

```

Activity #1 (Final Term Laboratory).py × a.txt main.py
1 from LinkedList import LinkedList as Stack, LinkedList
2 from LinkedList import LinkedList as Queue, LinkedList
3 from CircularQueue import CircularQueue as CircularQueue
4 from LinkedList import LinkedList as Deque
5 from PositionalList import PositionalList as PositionalList
6 Q = Queue()
7 S = Stack()
8 CircQ = CircularQueue()
9 D = Deque()
10 P = PositionalList()
11
12 from collections import deque
13 D = deque([1, 2, 3, 4, 5, 6, 7, 8])
14 Q = Queue()
15
16 D = deque([1, 2, 3, 4, 5, 6, 7, 8])
17 Q = LinkedList()
18
19 D.remove(4)
20 D.remove(5)
21 Q.enqueue(5)
22 Q.enqueue(4)
23
24 D.insert(_i: 3, Q.dequeue())
25 D.insert(_i: 4, Q.dequeue())
26
27 print("D after using LinkedList: ", list(D))
28
29 D = deque([1, 2, 3, 4, 5, 6, 7, 8])
30 S = LinkedList()

```

Activity #1 (Final Term Laboratory).py × main.py

```
16 from collections import deque
17 D = deque([1, 2, 3, 4, 5, 6, 7, 8])
18 Q = LinkedQueue() # Using custom LinkedQueue
19
20 D.remove(4)
21 D.remove(5)
22 Q.enqueue(5)
23 Q.enqueue(4)
24 D.insert(_i: 3, Q.dequeue())
25 D.insert(_i: 4, Q.dequeue())
26
27 print("D after using LinkedQueue: ", list(D))
28
29 D = deque([1, 2, 3, 4, 5, 6, 7, 8])
30 S = LinkedStack()
31 D.remove(4)
32 D.remove(5)
33 S.push(5)
34 S.push(4)
35
36 D.insert(_i: 3, S.pop())
37 D.insert(_i: 4, S.pop())
38
39 print("D after using LinkedStack: ", list(D))
40
```

"C:\Program Files\Python312\python.exe" "Z:\DSAL601-IDB2\FINALS\LinkedLists\Activity #1 (Final Term Laboratory).py"

D after using LinkedQueue: [1, 2, 3, 5, 4, 6, 7, 8]

D after using LinkedStack: [1, 2, 3, 4, 5, 6, 7, 8]

Process finished with exit code 0

```
1 from LinkedStack import LinkedStack as Stack
2 from LinkedQueue import LinkedQueue as Queue
3 from CircularQueue import CircularQueue as CircularQueue
4 from LinkedDeque import LinkedDeque as Deque
5 from PositionalList import PositionalList as PositionalList
6
7
8 Q = Queue()
9 S = Stack()
10 CircQ = CircularQueue()
11 D = Deque()
12 P = PositionalList()
13
14 # Insert elements into the deque using insert_last (since you are working with LinkedDeque)
15 D.insert_last(1)
16 D.insert_last(2)
17 D.insert_last(3)
18 D.insert_last(4)
19 D.insert_last(5)
20 D.insert_last(6)
21 D.insert_last(7)
22 D.insert_last(8)
23
24 # Perform operations on D using LinkedQueue
25 D.delete_last() # Removes 8
26 D.delete_last() # Removes 7
27
28 # Enqueue into the queue
29 Q.enqueue(5)
30 Q.enqueue(4)
31
```

```
31
32 # Insert dequeued elements into the deque
33 D.insert_first(Q.dequeue()) # Inserts 5 at the front of D
34 D.insert_first(Q.dequeue()) # Inserts 4 at the front of D
35
36 # Print the deque contents after using LinkedQueue
37 print("D after using LinkedQueue: ", D)
38
39 # Perform additional operations on D using LinkedStack
40 D.insert_first(1)
41 D.insert_first(2)
42 D.insert_first(3)
43 D.insert_first(4)
44 D.insert_first(5)
45 D.insert_first(6)
46 D.insert_first(7)
47 D.insert_first(8)
48
49 # Remove last two elements
50 D.delete_last() # Removes 1
51 D.delete_last() # Removes 2
52
53 # Push to stack
54 S.push(5)
55 S.push(4)
56
57 # Pop from stack and insert into deque
58 D.insert_first(S.pop()) # Inserts 5 at the front of D
59 D.insert_first(S.pop()) # Inserts 4 at the front of D
60
61 # Print the deque contents after using LinkedStack
62 print("D after using LinkedStack: ", D)
```