Carpio, Charlimagne V. Nov15,2024

DSALGO1-IDB2

```python
from collections import deque
new *
def enqueue(Q, e):
    Q.append(e)

new *
def dequeue(Q):
    if Q:
        return Q.pop(0)
    else:
        raise Exception("Queue is empty")

new *
def push(D, e):
    D.appendleft(e)

2 usages (2 dynamic)  new *
def pop(D):
    if D:
        return D.pop()
    else:
        raise Exception("Deque is empty")

D = deque([1, 2, 3, 4, 5, 6, 7, 8])
Q = []


four = D[3]
five = D[4]

D.remove(four)
D.remove(five)

D.insert( _i: 3, five)
D.insert( _i: 4, four)
```

```
new *
def push(D, e):
    D.appendleft(e)

2 usages (2 dynamic)  new *
def pop(D):
    if D:
        return D.pop()
    else:
        raise Exception("Deque is empty")

D = deque([1, 2, 3, 4, 5, 6, 7, 8])
Q = []

four = D[3]
five = D[4]

D.remove(four)
D.remove(five)

D.insert( _i: 3, five)
D.insert( _i: 4, four)

print("Dequeue:")
print("D:", list(D))
print("Q:", Q)
print()

print(D)

print()
```

Output

```
Dequeue:
D: [1, 2, 3, 5, 4, 6, 7, 8]
Q: []
```

```python
class Stack:
    new *
    def __init__(self):
        self.stack = deque()

    3 usages  new *
    def push(self, item):
        self.stack.append(item)

    1 usage  new *
    def pop(self):
        return self.stack.pop() if self.stack else None

    1 usage  new *
    def is_empty(self):
        return len(self.stack) == 0



D = deque([1, 2, 3, 4, 5, 6, 7, 8])
S = Stack()
```

```
for _ in range(3):
    S.push(D.popleft())


S.push(D.popleft())


S.push(D.popleft())


while not S.is_empty():
    D.appendleft(S.pop())



D[3], D[4] = D[4], D[3]

print("Stack:")
print(list(D))
```

Output:

```
Stack:
[1, 2, 3, 5, 4, 6, 7, 8]
```