queue-2

Assignment solution

PW SKILLS

1. Remove the last k elements of a queue.

Solution

```
from collections import deque

def remove_last_k_elements(n, k, elements):
    q = deque(elements)

    # Remove n - k elements from the front
    for _ in range(n - k):
        q.append(q.popleft())

    # Remove the last k elements
    for _ in range(k):
        q.pop()

    return list(q)

# Example usage:
n = 5
k = 3
elements = [1, 2, 3, 4, 5]

result = remove_last_k_elements(n, k, elements)
print(result)  # Output: [1, 2]
```

2. Reverse last k elements of a queue.

Solution

```
from collections import deque

def reverse_last_k_elements(n, k, elements):
    q = deque(elements)
    stack = []

    # Dequeue the first n-k elements and store them in a
stack
    for _ in range(n - k):
        stack.append(q.popleft())

    # Dequeue the last k elements and enqueue them back
after reversing
while q:
        stack.append(q.popleft())

    while stack:
        q.append(stack.pop())

    return list(q)

# Example usage:
n = 6
k = 3
elements = [1, 2, 3, 4, 5, 6]

result = reverse_last_k_elements(n, k, elements)
print(result)  # Output: [1, 2, 6, 5, 4, 3]
```

3. Implement queue using stacks [LeetCode 232]

Solution

```
class MyQueue:

    def __init__(self):
        self.input = []
        self.output = []

    def push(self, x: int) -> None:
        self.input.append(x)

    def pop(self) -> int:
        self.peek()
        return self.output.pop()

    def peek(self) -> int:
        if not self.output:
            while self.input:
                self.output.append(self.input.pop())
        return self.output[-1]

    def empty(self) -> bool:
        return not self.input and not self.output

# Example usage:
q = MyQueue()
q.push(1)
q.push(2)
print(q.peek())   # Output: 1
print(q.pop())    # Output: 1
print(q.empty())  # Output: False
```