Stack queue practise

Q1. Implement a stack and push 4 elements, then pop 2 elements.

Answer:

```
stack = []
stack.append(10)
stack.append(20)
stack.append(30)
stack.append(40)
print("Stack:", stack)       # [10, 20, 30, 40]

print("Popped:", stack.pop())  # 40
print("Popped:", stack.pop())  # 30
print("After Pops:", stack)    # [10, 20]
```

Q2. Check if the stack is empty after pushing and popping.

Answer:

```
stack = []
stack.append(5)
stack.pop()
print("Is empty?", not stack)   # True
```

Q3. Insert an element at the front of a list (queue).

Answer:

```
queue = ['A', 'B']
queue.insert(0, 'X')
print("After insert at front:", queue)  # ['X', 'A', 'B']
```

Q4. Create a queue, enqueue 3 elements, dequeue 1 element.

Answer:

```
queue = []
queue.append("apple")
queue.append("banana")
queue.append("cherry")
print("Queue:", queue)         # ['apple', 'banana',
'cherry']
```

```
print("Dequeued:", queue.pop(0)) # 'apple'
print("After dequeue:", queue)   # ['banana', 'cherry']
```

Q5. Use deque to add elements on both sides.

Answer:

```
from collections import deque
dq = deque()
dq.append(10)        # Right
dq.append(20)        # Right
dq.appendleft(5)     # Left
print("Deque:", dq)  # deque([5, 10, 20])
```

Q6. Use deque to remove elements from both sides.

Answer:

```
from collections import deque
dq = deque([5, 10, 20])
print("Pop right:", dq.pop())       # 20
print("Pop left:", dq.popleft())    # 5
print("After pops:", dq)            # deque([10])
```

Q7. Simulate browser back button using stack.

Answer:

```
stack = ["Home", "About", "Products", "Contact"]
print("Current page:", stack[-1])  # Contact
stack.pop()                         # Go back
print("After Back:", stack[-1])     # Products
```

If you want, I can now prepare **10 short exam-style Stack & Queue Q&A** in a **question paper format** so you can revise them quickly.
That way you can print and practice.