스택을 이용해 다음 연산을 지원하는 큐를 구현하라.

- void push(int x) Pushes element x to the back of the queue.
- int pop() Removes the element from the front of the queue and returns it.
- int peek() Returns the element at the front of the queue.
- boolean empty() Returns true if the queue is empty, false otherwise.

1.리스트

```
class MyQueue:

    def __init__(self):
        """
        Initialize your data structure here.
        """
        self.input = []
        self.output = []

    def push(self, x: int) -> None:
        """
        Push element x to the back of queue.
        """
        self.input.append(x)


    def pop(self) -> int:
        """
        Removes the element from in front of queue and returns that element.
        """
        self.peek()
        return self.output.pop()


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

    def empty(self) -> bool:
        """
        Returns whether the queue is empty.
        """
        return self.input == [] and self.output == []


# Your MyQueue object will be instantiated and called as such:
# obj = MyQueue()
```

```
# obj.push(x)
# param_2 = obj.pop()
# param_3 = obj.peek()
# param_4 = obj.empty()

class MyQueue:

    def __init__(self):
        """
        Initialize your data structure here.
        """
        self.s = []

    def push(self, x: int) -> None:
        """
        Push element x to the back of queue.
        """
        len_s = len(self.s)
        temp = []
        for _ in range(len_s):
            temp.append(self.s.pop())
        self.s.append(x)
        for _ in range(len_s):
            self.s.append(temp.pop())


    def pop(self) -> int:
        """
        Removes the element from in front of queue and returns that element.
        """
        return self.s.pop()


    def peek(self) -> int:
        """
        Get the front element.
        """
        return self.s[-1]

    def empty(self) -> bool:
        """
        Returns whether the queue is empty.
        """
        return len(self.s) == 0


# Your MyQueue object will be instantiated and called as such:
# obj = MyQueue()
# obj.push(x)
# param_2 = obj.pop()
```

```
# param_3 = obj.peek()
# param_4 = obj.empty()
```