Lab 4: Abstract Data Type Solution

2) Queues

1. Implement *Queue* class found in *myqueue.py*

```
 ...
 class Queue:
     """A first-in-first-out (FIFO) queue of items.

     Stores data in a first-in, first-out order. When removing an item from the
     queue, the least recently-added item (i.e. the oldest item in the Queue)
     is the one that is removed.
     # === Private Attributes ===
     # _items:
     #     The items stored in this queue. The front of the list represents
     #     the front of the queue.
     """
     _items: List
     def __init__(self) -> None:
         """Initialize a new empty queue."""
         self._items = []

     def is_empty(self) -> bool:
         """Return whether this queue contains no items.

         >>> q = Queue()
         >>> q.is_empty()
         True
         >>> q.enqueue(’hello’)
         >>> q.is_empty()
         False
         """
         return self._items == []

     def enqueue(self, item: Any) -> None:
         """Add <item> to the back of this queue.
         """
         self._items.append(item)

```

```python
    def dequeue(self) -> Optional[Any]:
        """Remove and return the item at the front of this queue.

        Return None if this Queue is empty.
        (We illustrate a different mechanism for handling an erroneous case.)

        >>> q = Queue()
        >>> q.enqueue('hello')
        >>> q.enqueue('goodbye')
        >>> q.dequeue()
        'hello'
        """
        if self.is_empty():
            raise EmptyStackError
        else:
            return self._items.pop(0)
...

```

Listing 1: task_2_q1_solution.py

2. Complete functions *product* and *product_star* in *myqueue.py*

```python
...
def product_star(integer_queue: Queue) -> int:
    """Return the product of integers in the queue.

    Precondition: integer_queue contains only integers.

    >>> primes = [2, 3, 5, 7, 11, 13, 17, 19, 23, 29]
    >>> prime_line = Queue()
    >>> for prime in primes:
    ...     prime_line.enqueue(prime)
    ...
    >>> product_star(prime_line)
    6469693230
    >>> prime_line.is_empty()
    False
    """
    side_queue = Queue()
    output = 1

    if integer_queue.is_empty():
        return 0

    # 1. Move elements from integer_queue to side_queue
    while not integer_queue.is_empty():
        dequeued_element = integer_queue.dequeue()

        # 1.1 While moving elements, multiply each of them to output.
        output *= dequeued_element
        side_queue.enqueue(dequeued_element)

    # 2. Move back elements from side_queue to integer_queue
    while not side_queue.is_empty():
        integer_queue.enqueue(side_queue.dequeue())

    return output
...

```

Listing 2: task_2_q2_solution.py

Notes:

- 오늘 형모 마음도 날씨도 봄 처럼 따뜻해요.
- 여보, 형모 사랑하는 내 여보 하고 손잡고 같이 걸을 수 있게 해줘서 고마워요 :)
- 형모 마음이 설레요
- 허허허허허허허
- 형모야. 오늘 무너지지 않고, 설레이는 마음 갖고, 차분히
- 화이팅 :)