**Tutorial 11**

**Queues**

1. Write out the content of the resulting queue after the following code is executed:

|  |
| --- |
| myQueue = Queue() **for** i **in** range( 16 ):  **if** i % 3 == 0:  myQueue.enqueue(i) |

a.

the contents of the queue:

[0]: 0 [1]: 3 [2]: 6 [3]: 9 [4]: 12 [5]: 15

|  |
| --- |
| myQueue = Queue() **for** i **in** range( 16 ): **if** i % 3 == 0:  myQueue.enqueue(i) **elif** i % 4 == 0:  myQueue.dequeue() |

b.

The contents of the queue:

[0]: 6 [1]: 9 [2]: 12 [3]: 15

|  |
| --- |
| myQueue = Queue() **for** i **in** range( 16 ): **if** i % 3 == 0:  myQueue.enqueue(i)  myQueue.enqueue(i + 1)  **elif** i % 4 == 0:  myQueue.dequeue() |

c.

IT2153/IT2352/IT2553/IT2653/IT2852

Data Structures & Algorithms



The contents of the queue:

[0]: 3 [1]: 4 [2]: 6 [3]: 7 [4]: 9 [5]: 10 [6]: 12 [7]: 13 [8]: 15 [9]: 16

|  |
| --- |
| myQueue = Queue()  **for** i **in** range( 16 ):  **if** i % 4 == 0:  myQueue.dequeue()  **elif** i % 3 == 0:  myQueue.enqueue(i) |

d.

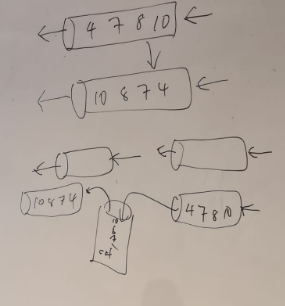
AssertionError: Cannot dequeue from an empty queue

1. Write a Python function – reverseQueue(Q) with no more than 6 lines of code, that reverses the order of the items in a queue. Your solution may only use the methods defined by the Queue ADT, but you are free to use other data structures if necessary

Graphical user interface, text, application

Description automatically generated with medium confidence

Diagram to help me understand better:



1. A double-ended queue, or **deque** (pronounced as “deck” to avoid confusion with the dequeue method of the regular Queue ADT) is a queue-like data structure that supports insertion and deletion at both the front and the back of the queue.

Define a Deque ADT by listing and describing the methods that should be supported by a Deque.

|  |  |
| --- | --- |
| Deque() | Creates a new empty deque, which is deque containing no items |
| isEmpty() | Returns a Boolean value indicating whether the dequeue is empty |
| length() | Returns the number of items currently in the deque |
| enqueue\_front(item) | Adds the given item to the front of the deque |
| enqueue\_back(item) | Adds the given item to the back of the deque |
| dequeue\_front() | Removes and returns the front item from the deque. An item cannot be dequeued from an empty deque |
| dequeue\_back() | Removes and returns the back item from the deque. An item cannot be dequeued from an empty deque. |

***-- End of Tutorial --***

AY2020/21 S1 Page 1