Faculty of
Information Technology
King Mongkut's Institute of Technology Ladkrabang

Name:.....

Student ID:.....

Assignment # 5: Queues

1. What is the main difference between a stack and a queue?

2. Give one example of real-world problems where a **circular queue** could be used. Then, provide details of how a circular queue is useful in the given example. Be careful to not provide an example that a normal queue also works well.

3. Given a class of Queue which is defined as follows:

```
class ArrayQueue:
  DEFAULT_CAPACITY = 10   # moderate capacity for all new queues

  def __init__(self):
    """ Create an empty queue with specified size. """
    self._data = [None] * ArrayQueue.DEFAULT_CAPACITY
    self._size = 0
    self._front = 0
    self._rear = 0

  def __len__(self):
    """ Return the number of elements in the queue. """
    return self._size

  def is_empty(self):
    """ Return True if the queue is empty. """
    return self._size == 0

  def first(self):

    """ Return (but do not remove) the element at the front of the queue.
    Raise Empty exception if the queue is empty."""
    if self.is_empty():
      raise Empty('Queue is empty')
    return self._data[self._front]

  def dequeue(self):

    """ Remove and return the first element of the queue.
    Raise Empty exception if the queue is empty."""
    if self.is_empty():
      raise Empty('Queue is empty')
    answer = self._data[self._front]
    self._data[self._front] = None          # Reclaiming unused space
    self._front = (self._front + 1) % len(self._data) # shift the location
of the front index rightward
    self._size -= 1
    return answer
```

```
  def enqueue(self, e):
    """ Add an element to the back of queue."""
    if self._size == len(self._data):
       self._resize(2*len(self._data))     # double the array size when all
slots are occupied.
    if self._rear == 0 and self._front == 0 and self._data[0] == None:
# For the first case of rear = 0 and no data in Q[0] - Empty Queue
      self._data[self._rear] = e
    else:
      self._rear = (self._rear + 1) % len(self._data) # shift the location
of the rear index rightward
      self._data[self._rear] = e
    self._size += 1         # Keep track of number of elements in ArrayQueue

  def _resize(self, cap):                # Assume cap >= len(self)
    """ Resize to a new list of capacity >= len(self). """

    old = self._data                     # keep track of existing list
    self._data = [None] * cap            # allocate list with new capacity
    walk = self._front
    for k in range(self._size):          # consider existing elements
      self._data[k] = old[walk]          # Shift indices to start at 0
      walk = (1 + walk) % len(old)       # use old size as modulus
    self._front = 0                      # front has been realigned
    self._rear = self._size - 1          # rear has been realigned
```

Given a queue class as implemented above, a method to create an empty queue is:

Q = ArrayQueue()

What values are returned during the following series of queue operations, if executed upon an initially empty queue?

Operation	Return Value	Queue values
Q.enqueue(7)		
Q.enqueue(2)		
Q.dequeue()		
Q.enqueue(5)		
Q.enqueue(9)		
Q.first()		
Q.dequeue()		
Q.dequeue()		
Q.is_empty()		
Q.enqueue(9)		
len(Q)		
Q.enqueue(0)		
Q.dequeue()		
Q.dequeue()		
Q.dequeue()		
Q.dequeue()		
len(Q)		

4. Based on a Queue class as defined in Question 3, write the pseudocodes or python codes for the following tasks.

4.1 Create 3 empty queues namely: A, B and C.

4.2 Add new items including “CPU”, “RAM”, “GPU”, “Mainboard”, “PSU”, and “SSD”, respectively, into queue “A”.

4.3 Add new items including “16 GB”, “2 TB”, “850 Watt”, “3.2 GHz”, “ATX”, and “1350 MHz”, respectively, into queue “B”.

4.4 Transfer the items from queue “A” and “B” to queue “C” such that the result of queue C is [‘CPU’, ‘3.2 GHz’, ‘Mainboard’, ATX’]’. Also, show the list elements of A and B after operations.

[Only use queue operations to move items]

Example use of method to transfer item from queue A to queue B:

```
B.enqueue(A.dequeue())
```

4.5 Rearrange items into the following results:

A = [‘RAM’, ‘16 GB’, ‘SSD’, ‘2 TB’],
B = [‘GPU’, ‘1350 Mhz’, ‘PSU’, ‘850 Watt’], and
C = [“CPU’, ‘3.2 GHz’, ‘Mainboard’, ATX”]

[Only use queue operations to move items]

[Removing all items in the queue then add items one by one is not allowed. Items can be only transferred from a queue to another queue.]

Example use of method to transfer item from queue A to queue B:

```
B.enqueue(A.dequeue())
```

4.6 Remove all items in Queue “C” by using a loop.

5. Suppose an initially empty queue Q has executed a total of 48 enqueue operations, 15 first operations, and 21 dequeue operations, 7 of which (i.e. dequeue) raised Empty errors that were caught and ignored.

What is the current size of Q? Also, show your calculation of how you obtain the size.