Course Title: Sequential and parallel Algorithm
Course Number & Section: COMP 273-01

Homework #1

Task #1:

Modify the **FixedCapacityQueueArray.java** code to resizable queue data structure that grows and shrinks based on the number of spaces in the queue. The resizing principle is:

- The queue should double its size when it is full.
- Shrinks its size when its size is quarter full.

Task #2:

Also, modify **FixedCapacityQueueArray.java** code to do the following:

- To accept any data type (i.e. making it to become a generic queue data structure)
- Enqueue characters and words from any given string of input

Hint:

1. You can leverage the split() of the string class for word enqueuing process.
2. Leverage the idea used in implementing resizable array stack in page 141 of the recommended textbook of this class.
3. The input format of the program is:

java QueueClient <enqueue type> <filename.txt> <num of dequeue op>

Either w=for word or c=for character

Expected Sample Output of the Resizable

- ## For character enqueue and dequeue operation

```
C:\Users\cosmo\OneDrive\Desktop\JAVADEMO\ArrayQueue\src>java QueueClient c Qstrin.txt 15
Size of queue before queue operation :1
Queue state after enqueuing item as Character
[I, h, a, v, e, a, d, r, e, a, m, o, f, b, e, c, o, m, i, n, g, a, c, o, m, p, u, t, e, r, g, e, e, k, !, null, null, null, null, null, null, null, null, nu
ll, null, null, null, null, null, null, null, null, null, null, null, null, null, null, null, null, null, null, null, null]:Size of queue is 64
Queue state after dequeuing item as Character
[null, null, null, null, null, null, null, null, null, null, null, null, null, null, null, c, o, m, i, n, g, a, c, o, m, p, u, t, e, r, g, e, e, k, !, null,
 null, null, null, null, null, null, null, null, null, null, null, null, null, null, null, null, null, null, null, null, null, null, null, null, null, null,
 null, null]:Size of queue is 64

C:\Users\cosmo\OneDrive\Desktop\JAVADEMO\ArrayQueue\src>java QueueClient c Qstrin.txt 25
Size of queue before queue operation :1
Queue state after enqueuing item as Character
[I, h, a, v, e, a, d, r, e, a, m, o, f, b, e, c, o, m, i, n, g, a, c, o, m, p, u, t, e, r, g, e, e, k, !, null, null, null, null, null, null, null, null, nu
ll, null, null, null, null, null, null, null, null, null, null, null, null, null, null, null, null, null, null, null, null]:Size of queue is 64
Queue state after dequeuing item as Character
[null, null, null, null, null, p, u, t, e, r, g, e, e, k, !, null, null, null, null, null, null, null, null, null, null, null, null, null, null, null, null,
 null]:Size of queue is 32
```

- ## For word enqueue and dequeue operation

```
C:\Users\cosmo\OneDrive\Desktop\JAVADEMO\ArrayQueue\src>java QueueClient w Qstrin.txt 5
Size of queue before queue operation :1
Queue state after enqueuing item as Word
[I, have, a, dream, of, becoming, a, computer, geek, !, null, null, null, null, null, null]:Size of queue is 16
Queue state after dequeuing item as Word
[null, null, null, null, null, becoming, a, computer, geek, !, null, null, null, null, null, null]:Size of queue is 16

C:\Users\cosmo\OneDrive\Desktop\JAVADEMO\ArrayQueue\src>java QueueClient w Qstrin.txt 8
Size of queue before queue operation :1
Queue state after enqueuing item as Word
[I, have, a, dream, of, becoming, a, computer, geek, !, null, null, null, null, null, null]:Size of queue is 16
Queue state after dequeuing item as Word
[null, geek, !, null, null, null, null, null]:Size of queue is 8
```