Lab 1 (Week 2)

The following source code represents the basic operations of the Queue data structure.

```
class Queue {
    constructor() {
        this.q = [];
    }
// get the current number of elements in the queue
//Getter function
    get length() {
        return this.q.length
    };
//Get all the elements
    get queue() {
        return this.q;
    }
// Boolean function: returns true if the queue is empty, false otherwise
    isEmpty() {
        return 0 == this.q.length;
    };
//adds new element to the end of the quue
    enqueue(newItem) {
        this.q.push(newItem)
    };
//Boolean function: returns true if an item is found (first occurnace); false otherwise
    inQueue(item) {
        let i = 0;
        let isFound = false;
        while (i < this.q.length && !isFound) {
```

```
            if (this.q[i] === item) {
                isFound = true;
            } else
                i++;
        }
        return (isFound);
    }
// pop an item from the queue
    dequeue() {
        if (0 != this.q.length) {
            let c = this.q[0];
            this.q.splice(0, 1);
            return c
        }
    };

};

let queue = new Queue();
queue.enqueue(10);
queue.enqueue(20);
console.log(queue.length);
console.log(queue.q);
queue.dequeue();
queue.enqueue(33);
console.log(queue.q);
console.log(queue.inQueue(33));
console.log(queue.inQueue(88));
```

Expected output

```
2
[ 10, 20 ]
[ 20, 33 ]
true
false
```

Tasks

- add a new function that removes all the elements in the queue
- add a new function that adds a set of items into the queue
  - E.g: queue.addAll([3,7,1,9])
- add a function that pops (dequeues) N elements from the queue. The function should reject the input if there is no enough element to be removed.
  - E.g: queue.dequeueN(2); // pop 2 elements
- add a new function that prints the content of the queue with their indexes. The output can be something like:
  - 1–>34
  - 2–>30
  - 3–>11
  - 4–>-3

Note: some text and codes have been taken from books and web pages.

References:

- https://en.wikipedia.org/wiki/MEAN_(software_bundle)
- https://developer.mozilla.org/en-US/docs/Web/JavaScript/Guide/
- http://www.dofactory.com/tutorial/javascript-loops
- https://javascript.info/

Copyright © Monash University, unless otherwise stated. All Rights Reserved, except for individual components (or items) marked with their own licence restrictions

Disclaimer and Copyright
Privacy
Service Status