CST 370
Homework (Queues)

1. (**20 points**) In the following code, assume the **myQueue** object is a queue that can hold integers. (The lines are numbered for reference purposes.)

```
1. myQueue.enqueue(200);
2. myQueue.enqueue(100);
3. myQueue.enqueue(300);
4. cout << myQueue.front() << endl;
5. myQueue.dequeue();
6. myQueue.dequeue();
7. cout << myQueue.front() << endl;
```

What will the statement in line 4 display? 200

What will the statement in line 7 display? 300

2. (**20 points**) Enqueue 5 numbers [6, 3, 1, -5, -10] in order. Then dequeue 3 elements from the queue. Print out contents of the current queue. -5, -10

3. (**40 points**) Write an algorithm (in pseudocode) to implement a stack using two queues (say q1 and q2). Specifically, you need to implement the pop() and push() functions of a stack. You can assume that you have the implementation of the queue available and you can use the enqueue() and dequeue() functions of the queue. Note stack is a LIFO data structure while queue is a FIFO data structure.

```
Class QueueStack

    private:
        Queue q1
        Queue q2
        int stackSize

    public:
         void push (QueueElement element)
        //input: queue
        //output: a queue with inputted element
            stackSize++
            q2.enqueue(element)

            while (!q2.element())
                q2.enqueue(q1.front())
                q1.dequeue()

            Queue q = q1
            q1 = q2
            q2 = q

        void pop()
        //input: queue
        //output: queue with one less element removed from the top
            if (q1.empty())
                return
            q1.dequeue()
            stackSize--
```