Hyo Lee
Week 3 Homework Submission
01/26/2016

CST 370
Homework (Stacks and Queues)

1. Convert the following infix expressions to postfix expressions
a) (2+3)*6+5*6-7 = **2 3 + 6 * 5 6 * + 7 -**
b) 2+3*7+(4-6*7) = **2 3 7 * + 4 6 7 * - +**

2. 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(100);
2. myQueue.enqueue(200);
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? **100**

What will the statement in line 7 display? **300**

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

4. Write an algorithm 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.

Though a pseudocode or code will be preferred you will still be given points if you describe the algorithm in plain English as a sequences of steps. For example, while writing the pseudocode if you want to call the enqueue() function for queue q1, you can call it as q1.enqueue().

```
class FakeStack
{
private:
	Queue q1, q2;

public:
	void push(QueueElement value)
	{
		if (q1.empty())
		{
			q1.enqueue(value);
		}
		else
		{
			while (!q1.empty())
			{
				q2.enqueue(q1.front());
				q1.dequeue();
			}
		}
		if (q1.empty())
		{
			q1.enqueue(value);
		}
		while (!q2.empty())
		{
			q1.enqueue(q2.front());
			q2.dequeue();
		}
	}

	void pop()
	{
		cout << q1.front() << " ";
		q1.dequeue();
	}
};


int main()
{
	FakeStack s;
	s.push(1);
	s.pop();
}
```