Matt Muldowney
Homework 7 due 4/16

Written Problems

Problem 1

```
typedef struct queue {
	int data[MAX_LEN];
	int end;
}

void enqueue(queue* q, int val) {
	int loc;
	for (int i = q->end-1; q->data[i] >= val; i--) {
		q->data[i+1] = q->data[i];
		loc = i;
	}

	q->data[loc] = val;
	q->end++;
}

int dequeue(queue* q) {
	return q->data[q->(--end)];
}
```

When enqueueing an element, we want to insert into the array so that the array remains sorted (from lowest priority to highest priority). To do this, shift over all elements of equal or greater priority and the insert the element into the array. This has **space cost O(1)** and **time cost O(n)**. Because the array is always sorted, the highest priority

element will always be at the end of the array. So when dequeueing, we simply process the element at the last index. This has **space cost O(1)** and **time cost O(1).**