CS 106B Section 2 (Week 3) Solutions

---

1. reorder

```
void reorder(Queue<int>& q) {
    Stack<int> s;
    int size = q.size();

    // Separate positive and negative numbers
    for (int i = 0; i < size; i++) {
        int n = q.dequeue();
        if (n < 0) {
            s.push(n);
        } else {
            q.enqueue(n);
        }
    }

    // Enqueue negative numbers in reverse order
    size = q.size();
    while (!s.isEmpty()) {
        q.enqueue(s.pop());
    }

    // Move positive numbers to end of queue
    for (int i = 0; i < size; i++) {
        q.enqueue(q.dequeue());
    }
}
```

---

2. twice

```
Set<int> twice(Vector<int>& v) {
    Map<int, int> counts;
    for (int i : v) {
        counts[i]++;
    }
    Set<int> twice;
    for (int i : counts) {
        if (counts[i] == 2) {
            twice += i;
        }
    }
    return twice;
}
```

Bonus solution:

```
Set<int> twice(Vector<int>& v) {
    Set<int> once;
    Set<int> twice;
    Set<int> more;
    for (int i : v) {
        if (once.contains(i)) {
            once.remove(i);
            twice.add(i);
        } else if (twice.contains(i)) {
            twice.remove(i);
            more.add(i);
        } else if (!more.contains(i)) {
            once.add(i);
        }
    }
    return twice;
}
```

CS 106B Section 2 (Week 3) Solutions

---

3. unionSets

```
Set<int> unionSets(HashSet<Set<int> >& sets) {
    Set<int> all;
    for (Set<int> s : sets) {
        all += s;
    }
    return all;
}
```

---

4. reverse *(map)*

```
Map<string, int> reverse(Map<int, string>& map) {
    Map<string, int> rev;
    for (int i : map) {
        rev[map[i]] = i;
    }
    return rev;
}
```

---

5. print2grams

```
void print2grams(Map<string, Map<string, double> >& twoGrams) {
    for (string first : twoGrams) {
        for (string second : twoGrams[first]) {
            cout << first << " " << second << ": "
                 << twoGrams[first][second] << endl;
        }
    }
}
```

---

6. mystery *(recursion I)*

```
1 => 1
15 => 6
314 => 8
271828 => 28
-1414 => 10
```

---

7. cannonballs *(recursion II)*

```
int cannonballs(int height) {
    if (height == 0) {
        return 0;
    } else {
        return height * height + cannonballs(height - 1);
    }
}
```

---

8. reverse *(recursion III)*

```
string reverse(string s) {
    if (s == "") {
        return "";
    } else {
        return reverse(s.substr(1)) + s[0];
    }
}
```