Section #2

Tyler Conklin and Kate Rydberg (Based on handouts by previous CS106B instructors and TAs, especially those of Ashley Taylor and Shreya Shankar.)

File I/O, Stacks, Queues: Solutions

1. Input Stats (file I/O)

```
void inputStats(string filename) {
    ifstream input;
    input.open(filename);
    if (input.fail()) {
        return;
    }
    int lineCount = 0;
    int longest = 0;
    int totalChars = 0;
    string line;
    while (getline(input, line)) {
        lineCount++;
        totalChars += line.length();
        longest = max(longest, (int) line.length());
        cout << "Line " << lineCount << " has " << line.length() << " chars" << endl;
    }
    double average = (double) totalChars / lineCount;
    cout << lineCount << " lines; longest = " << longest
         << ", average = " << average << endl;
}
```

2. Collections Mystery (Stack and Queue)

1. {**6, 5, 4, 2, 1, 2, 3**}

2. {**90, 71, 84, 67, 29, 115, 84, 33**}

3. Reorder (Queue)

```
void reorder(Queue<int>& q){
    Stack<int> s;
    int oldSize = q.size();
    for (int i = 0; i < oldSize; i++) {
        int n = q.dequeue();
        if (n < 0) {
            s.push(n);
        } else {
            q.enqueue(n);
        }
    }
    int newSize = q.size();
    while (!s.isEmpty()) {
        q.enqueue(s.pop());
    }
    for (int i = 0; i < newSize; i++) {
        q.enqueue(q.dequeue());
    }
}
```

4. Check Balance (Stack)

```
int checkBalance(string code) {
    Stack<char> parens;
    for (int i = 0; i < (int) code.length(); i++) {
        char c = code[i];
        if (c == '(' || c == '{') {
            parens.push(c);
        } else if (c == ')' || c == '}') {
            if (parens.isEmpty()) {
                return i;
            }
            char top = parens.pop();
            if ((top == '(' && c != ')') || (top == '{' && c != '}')) {
                return i;
            }
        }
    }
    if (parens.isEmpty()) {
        return -1;            // balanced
    } else {
        return code.length();
    }
}
```

5. Big-O

$\mathbf{O(n^2)}$

6. Maps and Sets

```
Map<string, Set<string>> friendList(string filename) {
    string s1, s2;
    ifstream infile;
    infile.open(filename);
    Map<string, Set<string>> friends;
    while (infile >> s1 >> s2) {
        friends[s1].add(s2);
        friends[s2].add(s1);
    }
    return friends;
}
```