CS 106B Section 3 (Week 4) Solutions

1. writeChars

```
void writeChars(int n) {
    if (n < 1) {
        throw "Invalid input.";
    } else if (n == 1) {
        cout << "*";
    } else if (n == 2) {
        cout << "**";
    } else {
        cout << "<";
        writeChars(n - 2);
        cout << ">";
    }
}
```

2. isMeasurable *(logic: explore all options – don't add weight, add weight to left, or add weight to right)*

```
bool isMeasurable(int target, Vector<int>& weights) {
    if (weights.isEmpty()) {
        return target == 0;
    }
    int first = weights[0];
    Vector<int> rest = weights; // need to use a copy!
    rest.remove(0);
    return isMeasurable(target, rest)
        || isMeasurable(target - first, rest)
        || isMeasurable(target + first, rest);
}
```

3. waysToClimb *(logic: explore all step combinations where at any step, you can either take 1 step or take 2 steps)*

```
int waysToClimb(int steps) {
    if (steps <= 0) {
        throw "Invalid input.";
    } else if (steps <= 2) {
        return steps; // 1 way to climb 1 step and
                      // 2 ways to climb 2 steps
    } else {
        return waysToClimb(steps - 1) +
                waysToClimb(steps - 2);
    }
}
```

4. isSubsequence *(logic: compare letters until you find every letter in **small** or exhaust the letters in **big**)*

```
bool isSubsequence(string big, string small) {
    if (small == "") {
        return true;
    } else if (big == "") {
        return false;
    } else {
        if (big[0] == small[0]) {
            return isSubsequence(big.substr(1), small.substr(1));
        } else {
            return isSubsequence(big.substr(1), small);
        }
    }
}
```

5. Debugging

The middle index is repeated in both sub-ranges, so when the recursion gets down to a range of length 2, the recursive call doesn't actually get any smaller, so it loops infinitely. Fix: replace the second recursive call with recursiveMax(v, middle + 1, right).