CS106B Aut 19 10/07/2019

Week 3 Section

1. RecursionMysteryComma - Recursion Trace

	Output:
`recursionMysteryComma(4, 1);`	4
`recursionMysteryComma(4, 2);`	8, 4, 8
`recursionMysteryComma(8, 2);`	16, 8, 16
`recursionMysteryComma(4, 3);`	12, 8, 4, 8, 12
`recursionMysteryComma(3, 4);`	12, 9, 6, 3, 6, 9, 12

2. sumOfSquares - Recursion

```
int sumOfSquares(int n) {
    if (n < 0) {
        throw n;
    } else if (n == 0) {
        return 0;
    } else {
        return n * n + sumOfSquares(n - 1);
    }
}
```

3. Zigzag - Recursion

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

Week 3 Section

4. StutterStack - Recursion

```
void stutterStack(Stack<int>& s) {
    if (!s.isEmpty()) {
        int n = s.pop();
        stutterStack(s);
        s.push(n);
        s.push(n);
    }
}
```

5. IsSubsequence - Recursion

```
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);
        }
    }
}
```

6. reverseLines - Recursion, File I/O

```
    void reverseLines(ifstream& input) {
       string line;
       if (getline(input, line)) {
           // recursive case
           reverseLines(input);
           cout << line << endl;
       } // else implicit base case, do nothing
    }
```