```
1 #include <iostream>
 2 #include <queue>
 3 #include <set>
 4 #include <stack>
 5 using namespace std;
 6
7 queue<int> reverseQueue(queue<int> input) {
 8
       queue<int> result;
 9
       stack<int> temp;
10
       while(!input.empty()) {
11
12
           temp.push(input.front());
           input.pop();
13
14
       }
15
       while(!temp.empty()) {
           result.push(temp.top());
16
17
           temp.pop();
18
       }
       return result;
19
20 }
21
22 set<int> makeUnion(set<int> s1, set<int> s2) {
       set<int> result;
23
24
       set<int>::iterator iter = s1.begin();
25
26
       while(iter != s1.end()) {
27
           result.insert(*iter);
28
           iter++;
       }
29
30
       set<int>::iterator itera = s2.begin();
       while(itera != s2.end()) {
31
32
           result.insert(*itera);
33
           itera++;
34
       }
35
       return result;
36 }
37
38 stack<int> removeNegative(stack<int> input) {
```

```
39
       stack<int> result, temp;
40
       while(!input.empty()) {
41
            if(input.top() >= 0)
42
43
                temp.push(input.top());
44
           input.pop();
45
       }
46
       while(!temp.empty()) {
           result.push(temp.top());
47
48
           temp.pop();
       }
49
50
51
       return result;
52 }
53
54
55 int main()
56 {
57
       std::cout << "Hello, World!" << std::endl;</pre>
58
59
       cout << endl << "Problem 0" << endl;</pre>
60
       queue<int> q;
       q.push(1); //gets food first
61
62
       q.push(2);
       q.push(3); //gets food last. can't get here
63
   without getting to the preceding ones
64
65
       queue<int> reversedQ = reverseQueue(q);
       while(!reversedQ.empty()) {
66
67
            cout << reversedQ.front() << endl;</pre>
68
            reversedQ.pop();
       }
69
70
       cout << endl << "Problem 1" << endl;</pre>
71
72
       set<int> s1, s2;
       s1.insert(1);
73
74
       s1.insert(2);
75
       s2.insert(1);
```

```
s2.insert(3);
 76
 77
        s2.insert(4);
 78
 79
        set<int> unionized = makeUnion(s1, s2);
 80
        set<int>::iterator iter = unionized.begin();
 81
        while(iter != unionized.end()) {
             cout << *iter << endl;</pre>
 82
 83
             iter++;
        }
 84
 85
 86
        cout << endl << "Problem 2" << endl;</pre>
 87
 88
        stack<int> st;
 89
        st.push(1);
 90
        st.push(-1);
 91
        st.push(-11);
 92
        st.push(10);
 93
 94
        stack<int> posS = removeNegative(st);
 95
        while(!posS.empty()) {
             cout << posS.top() << endl;</pre>
 96
 97
             posS.pop();
        }
 98
 99
100
        return 0;
101 }
102
```