

```
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
11     while(!input.empty()) {
12         temp.push(input.front());
13         input.pop();
14     }
15     while(!temp.empty()) {
16         result.push(temp.top());
17         temp.pop();
18     }
19     return result;
20 }
21
22 set<int> makeUnion(set<int> s1, set<int> s2) {
23     set<int> result;
24
25     set<int>::iterator iter = s1.begin();
26     while(iter != s1.end()) {
27         result.insert(*iter);
28         iter++;
29     }
30     set<int>::iterator itera = s2.begin();
31     while(itera != s2.end()) {
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
41     while(!input.empty()) {
42         if(input.top() >= 0)
43             temp.push(input.top());
44         input.pop();
45     }
46     while(!temp.empty()) {
47         result.push(temp.top());
48         temp.pop();
49     }
50
51     return result;
52 }
53
54
55 int main()
56 {
57     std::cout << "Hello, World!" << std::endl;
58
59     cout << endl << "Problem 0" << endl;
60     queue<int> q;
61     q.push(1); //gets food first
62     q.push(2);
63     q.push(3); //gets food last. can't get here
        without getting to the preceding ones
64
65     queue<int> reversedQ = reverseQueue(q);
66     while(!reversedQ.empty()) {
67         cout << reversedQ.front() << endl;
68         reversedQ.pop();
69     }
70
71     cout << endl << "Problem 1" << endl;
72     set<int> s1, s2;
73     s1.insert(1);
74     s1.insert(2);
75     s2.insert(1);
```

```
76     s2.insert(3);
77     s2.insert(4);
78
79     set<int> unionized = makeUnion(s1, s2);
80     set<int>::iterator iter = unionized.begin();
81     while(iter != unionized.end()) {
82         cout << *iter << endl;
83         iter++;
84     }
85
86     cout << endl << "Problem 2" << endl;
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()) {
96         cout << posS.top() << endl;
97         posS.pop();
98     }
99
100    return 0;
101 }
102
```