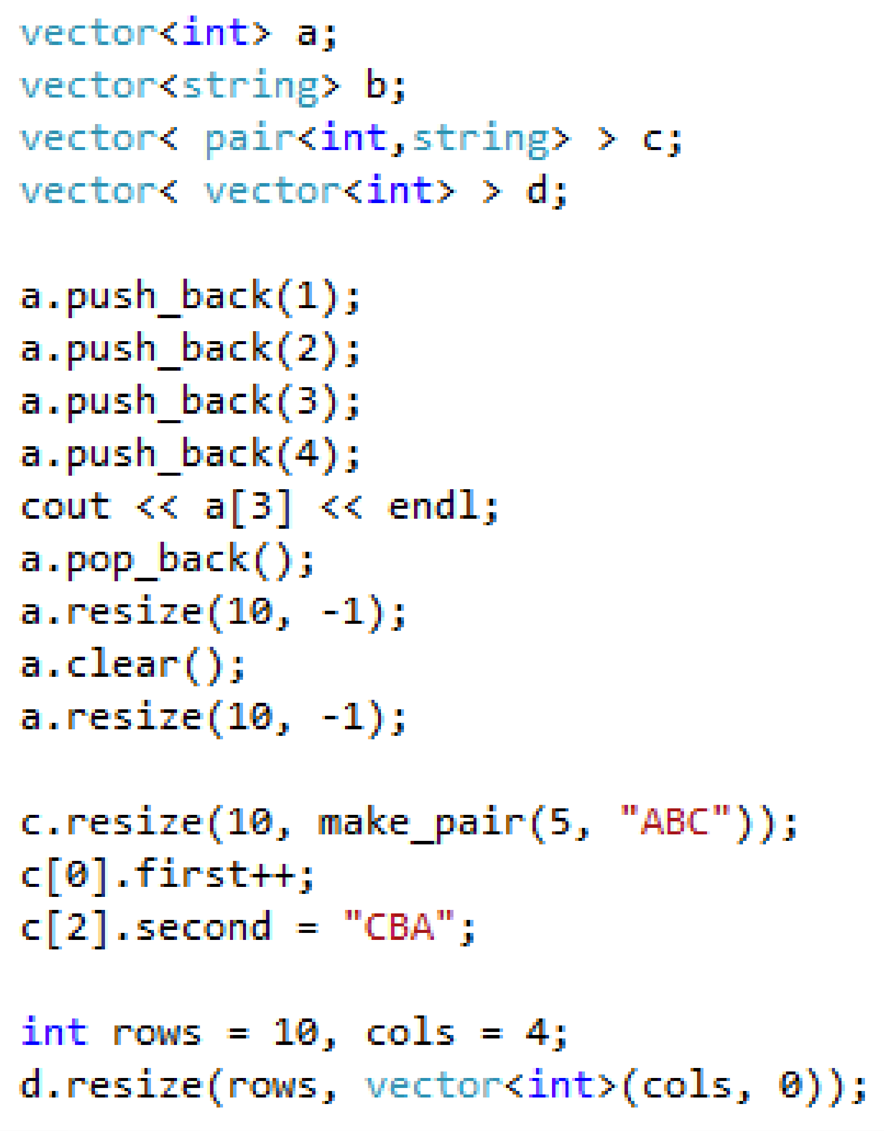
STL - vector



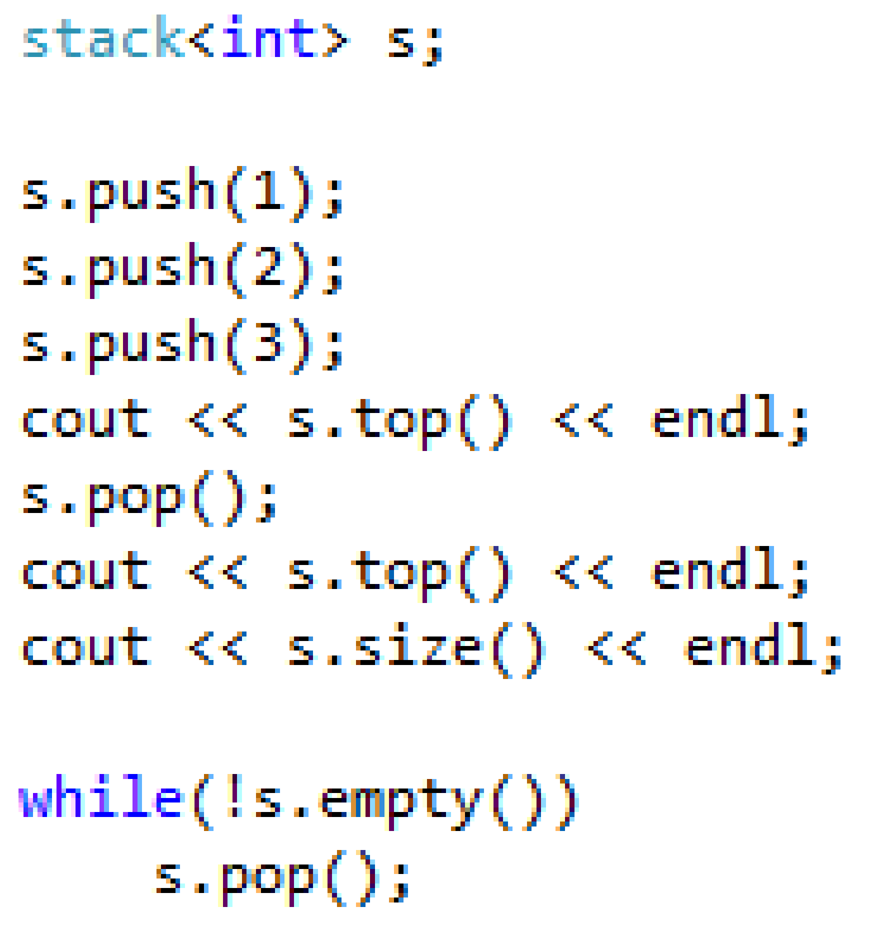
```
vector<int> a;
vector<string> b;
vector< pair<int,string> > c;
vector< vector<int> > d;

a.push_back(1);
a.push_back(2);
a.push_back(3);
a.push_back(4);
cout << a[3] << endl;
a.pop_back();
a.resize(10, -1);
a.clear();
a.resize(10, -1);

c.resize(10, make_pair(5, "ABC"));
c[0].first++;
c[2].second = "CBA";

int rows = 10, cols = 4;
d.resize(rows, vector<int>(cols, 0));
```

STL - stack

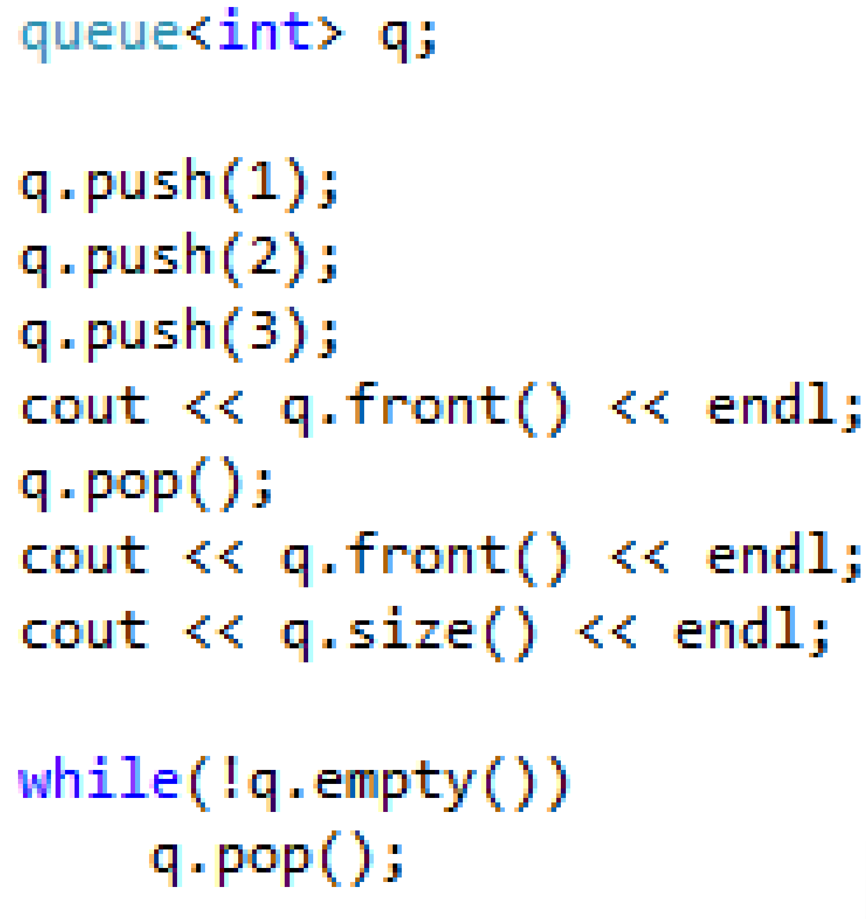


```
stack<int> s;

s.push(1);
s.push(2);
s.push(3);
cout << s.top() << endl;
s.pop();
cout << s.top() << endl;
cout << s.size() << endl;

while(!s.empty())
    s.pop();
```

STL - queue

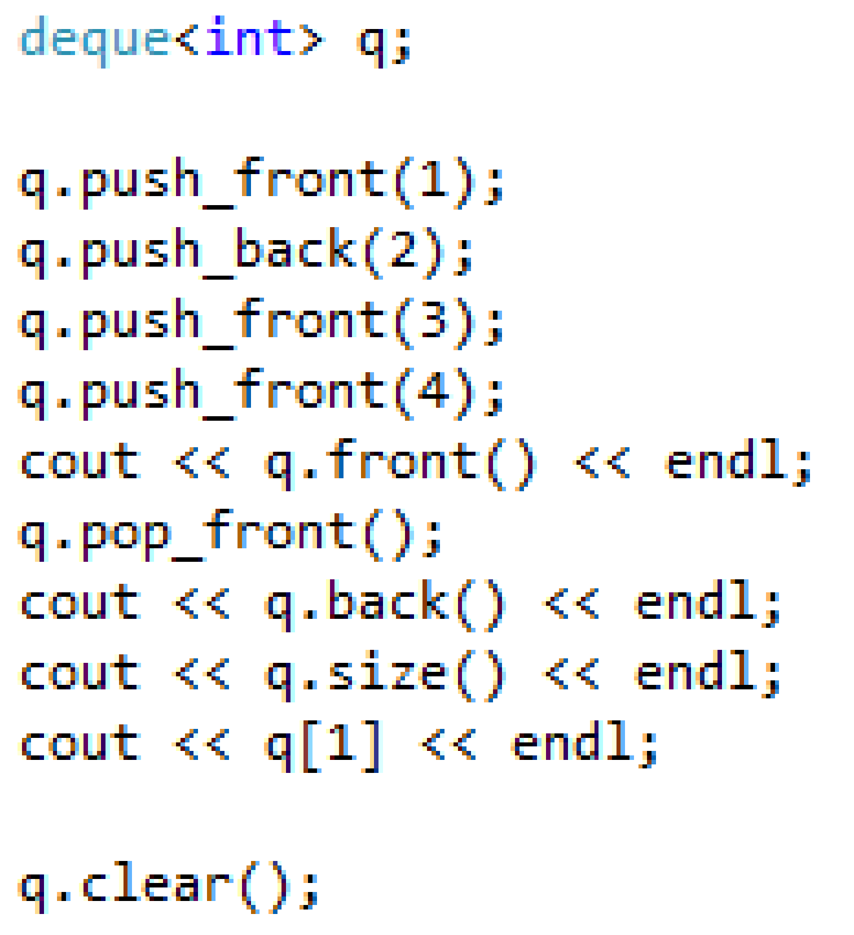


```
queue<int> q;

q.push(1);
q.push(2);
q.push(3);
cout << q.front() << endl;
q.pop();
cout << q.front() << endl;
cout << q.size() << endl;

while(!q.empty())
    q.pop();
```

STL - deque

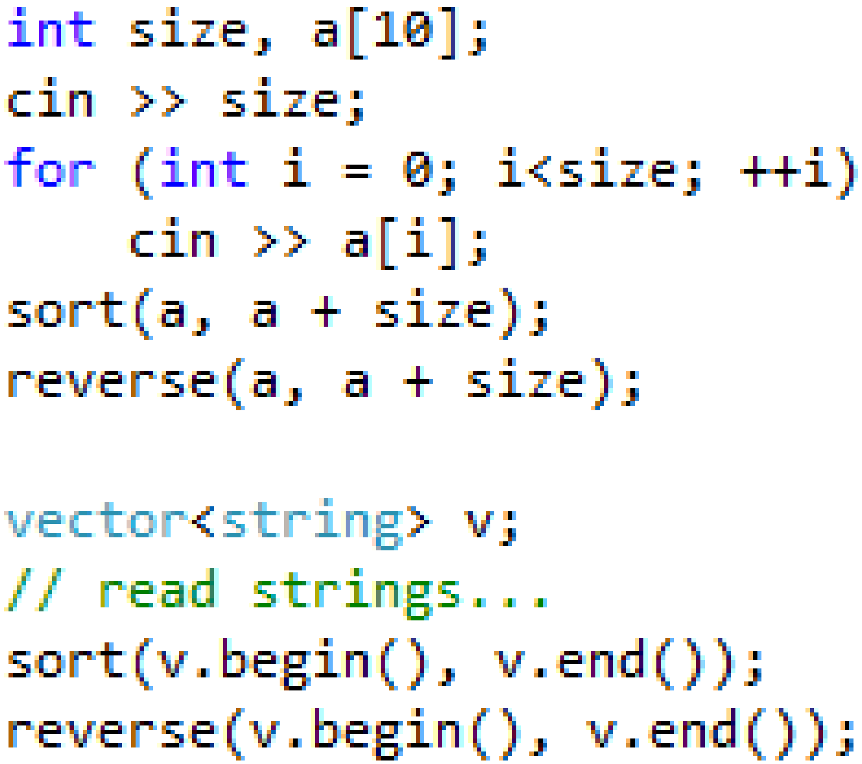


```
deque<int> q;

q.push_front(1);
q.push_back(2);
q.push_front(3);
q.push_front(4);
cout << q.front() << endl;
q.pop_front();
cout << q.back() << endl;
cout << q.size() << endl;
cout << q[1] << endl;

q.clear();
```

STL - sort & reverse



```
int size, a[10];
cin >> size;
for (int i = 0; i<size; ++i)
    cin >> a[i];
sort(a, a + size);
reverse(a, a + size);

vector<string> v;
// read strings...
sort(v.begin(), v.end());
reverse(v.begin(), v.end());
```

Problem #1

Given N numbers, print the given numbers in non-decreasing order.

Input:

```
5
3 2 4 3 1
```

Output:

```
1 2 3 3 4
```

Problem #2

http://codeforces.com/contest/81/problem/A

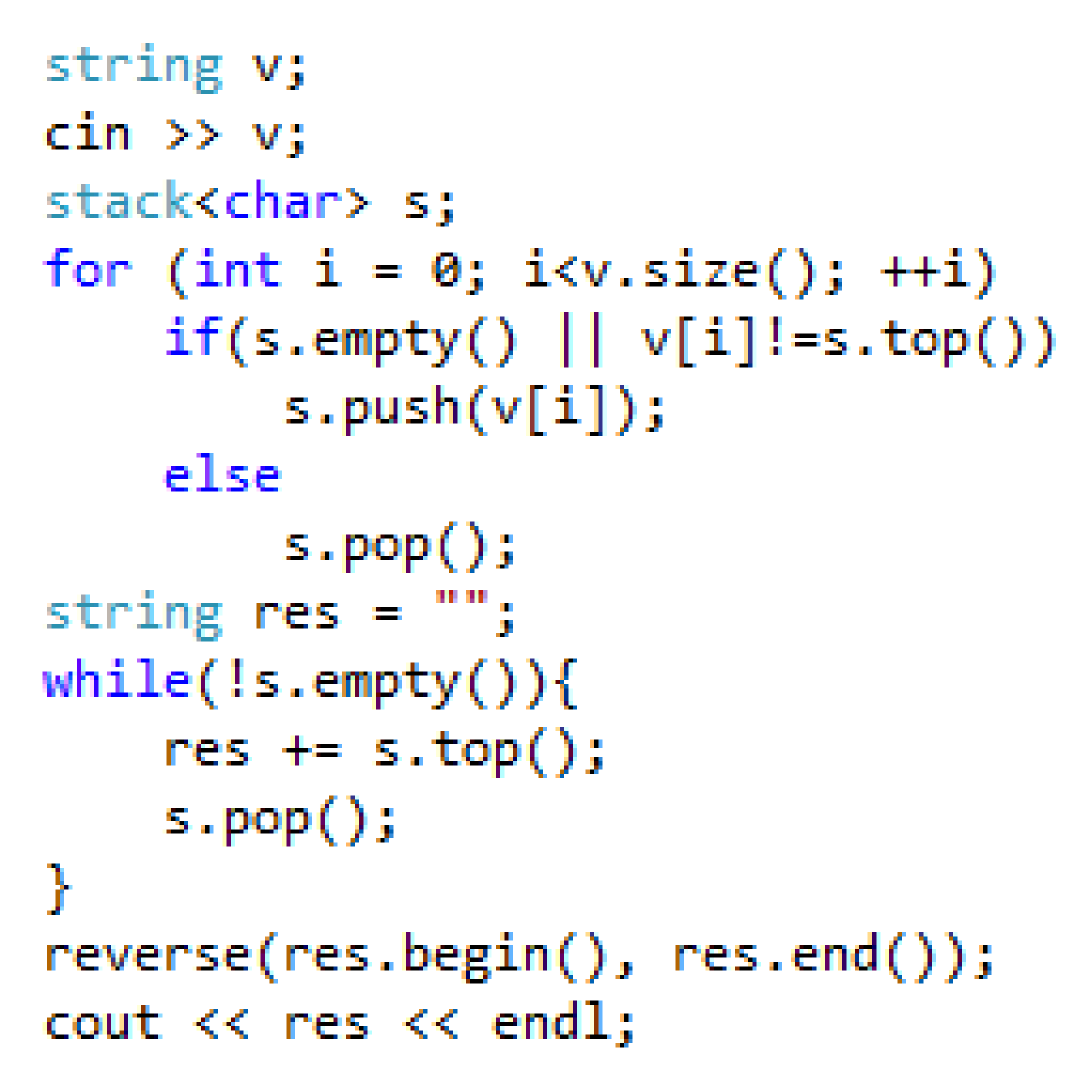
Input

```
hhoowaaaareyyoouu
```

Output

```
wre
```

Problem #2



```
string v;
cin >> v;
stack<char> s;
for (int i = 0; i<v.size(); ++i)
    if(s.empty() || v[i]!=s.top())
        s.push(v[i]);
    else
        s.pop();
string res = "";
while(!s.empty()){
    res += s.top();
    s.pop();
}
reverse(res.begin(), res.end());
cout << res << endl;
```

Problem #3

http://codeforces.com/contest/96/problem/B

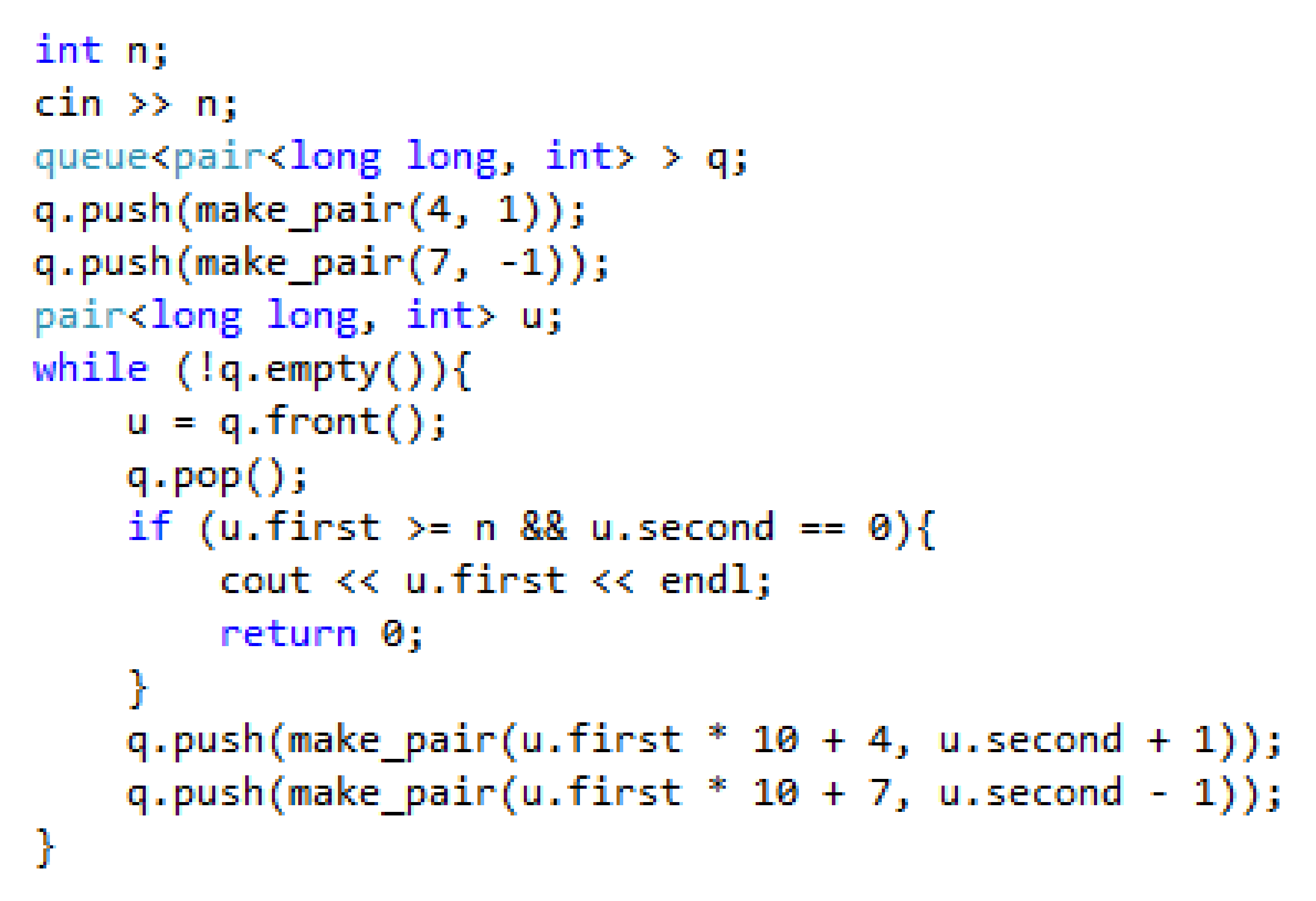
Input

```
4500
```

Output

```
4747
```

Problem #3



```
int n;
cin >> n;
queue<pair<long long, int> > q;
q.push(make_pair(4, 1));
q.push(make_pair(7, -1));
pair<long long, int> u;
while (!q.empty()){
    u = q.front();
    q.pop();
    if (u.first >= n && u.second == 0){
        cout << u.first << endl;
        return 0;
    }
    q.push(make_pair(u.first * 10 + 4, u.second + 1));
    q.push(make_pair(u.first * 10 + 7, u.second - 1));
}
```

References

- http://www.cplusplus.com/reference/