#include <iostream>

#include <stack>

#include <queue>

std::stack<int> reverse\_stack(std::stack<int> s) {

std::stack<int> reversed\_s;

//for(int i=0;i<s.size();i++)

while (!s.empty()) {

reversed\_s.push(s.top());

s.pop();

}

// write code here that returns a stack whose elements are

// in reverse order from those in stack s

return reversed\_s;

}

std::queue<int> reverse\_queue(std::queue<int> q) {

std::queue<int> reversed\_q;

std::stack<int> st;

while (!q.empty()) {

st.push(q.front());

q.pop();

}

while (!st.empty()) {reversed\_q.push(st.top());

st.pop();

}

// write code here that returns a queue whose elements are

// in reverse order from those in queue q

return reversed\_q;

}

void print\_stack(std::string name, std::stack<int> s) {

std::cout << "stack " << name << ": ";

while (!s.empty()) {

std::cout << s.top() << " ";

s.pop();

}

std::cout << std::endl;

}

void print\_queue(std::string name, std::queue<int> q) {

std::cout << "queue " << name << ": ";

while (!q.empty()) {

std::cout << q.front() << " ";

q.pop();

}

std::cout << std::endl;

}

int main() {

std::stack<int> s, rs;

std::queue<int> q, rq;

s.push(1); s.push(2); s.push(3); s.push(4); s.push(5);

print\_stack("s",s);

rs = reverse\_stack(s);

print\_stack("reversed s",rs);

q.push(1); q.push(2); q.push(3); q.push(4); q.push(5);

print\_queue("q",q);

rq = reverse\_queue(q);

print\_queue("reversed q",rq);

return 0;

}