Congratulations! You passed!

TO PASS 80% or higher

Keep Learning

GRADE
100%

Week 1 Challenge

LATEST SUBMISSION GRADE
100%

1. **Write functions that reverse the elements in a stack and in a queue. The starter code below include the STL <stack> and <queue> data structures.**

   A stack of integers is declared as "std::stack<int>" and the stack's top() member function returns the integer at the top of the stack (but also leaves it at the top of the stack). The push() method pushes a new integer onto the top of the stack and the pop() method deletes the value at the top of the stack.

   A queue of integers is declared as "std::queue<int>" and the queue's front() member function returns the integer at the front of the queue (but also leaves it at the front of the queue). The push() method pushes a new integer onto the back of the queue and the pop() method deletes the value at the front of the queue.

   Your job is to implement procedures that reverse the order of elements in a stack, and in a queue. The procedures print_stack() and print_queue() are provided to help you see if your procedures work.

   5 / 5 points

```cpp
#include <iostream>
#include <stack>
#include <queue>

std::stack<int> reverse_stack(std::stack<int> s) {
  std::stack<int> reversed_s;

  // write code here that returns a stack whose elements are
  // in reverse order from those in stack s
  while( 0 != s.size() )
  {
    int val = s.top();
    s.pop();
    reversed_s.push( val );
  }


  return reversed_s;
}

std::queue<int> reverse_queue(std::queue<int> q) {
  std::queue<int> reversed_q;
  std::stack<int> stk;
  // write code here that returns a queue whose elements are
  // in reverse order from those in queue q

  while( 0 != q.size() )
  {
    int val = q.front();
    q.pop();
    stk.push( val );
  }

  while( 0 != stk.size() )
  {
    int val = stk.top();
    stk.pop();
    reversed_q.push( val );
  }

  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;
}
```

Run

Reset

Correct

```
stack sent: 93 15 77 86 83
stack returned: 83 86 77 15 93
queue sent: 35 86 92 49 21
queue returned: 21 49 92 86 35
```