Quiz • 30 min

Week 1 Challenge

✓
Submit your assignment
DueMay 25, 12:29 PM IST
Try again
✓
Receive grade
To Pass80% or higher
Grade
100%
View Feedback
We keep your highest score

Confirm Navigation

Are you sure you want to leave this page?
Stay on this Page Leave this Page

Confirm Navigation

Are you sure you want to leave this page?
Stay on this Page Leave this Page

Week 1 Challenge

Graded Quiz • 30 min

Due May 25, 12:29 PM IST

✓

Congratulations! You passed!
To Pass 80% or higher
Keep Learning
Grade
100%

Week 1 Challenge

Latest Submission Grade
100%

1.Question 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.

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

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


```

```cpp
  for(int i=0;i<5;i++)
  {
  reversed_s.push(s.top());
  s.pop();
  }


      return reversed_s;
}

std::queue<int> reverse_queue(std::queue<int> q) {
  std::queue<int> reversed_q;
   int s = q.size();

    // Second queue


    for (int i = 0; i < s; i++) {

        // Get the last element to the
        // front of queue
        for (int j = 0; j < q.size() - 1; j++) {
            int x = q.front();
            q.pop();
            q.push(x);
        }

        // Get the last element and
        // add it to the new queue
       reversed_q.push(q.front());
        q.pop();
    }


  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
```

5 / 5 points