C++ STL Stack, Queue, and Deque

- C++ STL stack and queue containers
- Problems
  - Checking if symbols are balanced
  - Checking if input strings are palindrome
  - A programming contest problem (steps)

- C++ STL deque container
- Word puzzle problem

- All examples are in Examples/r5

C++ STL Stack

- stack(): zero parameter default constructor
- stack(const Container & con): one parameter constructor
- empty(): check if the stack is empty
- push(cont T& val): push value val into the stack
- pop(): delete the top element from the stack
- size(): return the number of elements in stack
- top(): return a reference to the top elements in stack

- More information
  - http://www.cppreference.com/wiki/stl/stack/start

C++ STL Queue

- queue(): zero parameter default constructor
- queue(const Container & con): one parameter constructor
- empty(): check if the queue is empty
- push(cont T& val): add element val to the end of the queue
- pop(): delete the front element from the queue
- size(): return the number of elements in queue
- front(): return a reference to the front element in queue
- back(): return reference to the last element in queue
-
- More information
  - http://www.cppreference.com/wiki/stl/queue/start

P1: Checking If Symbols are Balanced

- Problem description (balancingsymbols.pdf)
- Outline of the solution (using stack)

```
while (has next symbol)
        if next symbol is [ or (
                stk.push(symbol);
        else if next symbol is ] or )
                if stk.top() == symbol
                        stk.pop()
                else
                        not balanced
end while
if (!stk.empty())
        not balanced
```

- Source code (which contains a run-time error)
  - examples/r5/balancingsymbols_wrong.cpp
  - Example input file: balancingsymbols.input
  - Demo how to debug this program using gdb with coredump file

P2: Checking if Input Strings are Palindrome

- Problem statement (palindrome.pdf)
- Basic idea
  - Using a queue and stack to save input string in both queue and stack
  - Pop out characters from queue and stack and compare them one by one
  - Note that queue provides the forward reading, while stack provides the backward reading
  - Palindrome if characters from queue and stack are all equal
- Source code
  - examples/r5/palindrome.cpp
  - Example input file: palindrome.input

Palindrome: Recursive Version

- How to recursively determine if a string is palindrome?

- See examples/r5/palindrome_recursive.cpp

Steps

- This is a programming contest problem
  - A bit more complicated than our normal problems
  - But you can see the usages of STL containers and recursive algorithms
- Problem statement
  - steps.pdf
- Solutions
  - steps_stack.cpp (using stack to maintain the steps)
  - steps_recursive.cpp (a recursive solution, note how we pass parameter total)
- Test case
  - steps_input.txt
  - steps_output.txt
- To compile
  - Make steps_stack.x (or make steps_recursive.x)

STL Deque Container

- Header file <deque>
- Important member functions
  - begin() and end(): return iterators
  - front() and back(): reference to first and last elements
  - clear(): delete all elements
  - push_front(): insert at front
  - push_back(): insert at end
  - pop_front(): delete first element
  - pop_back(): delete last element
  - size(): number of elements
  - empty(): if vector is emtpy
  - resize(): change vector size
- More information at
  - http://www.cplusplus.com/reference/deque/deque/

Word Puzzle

- Problem description
  - word_puzzle.pdf
- Explaining the source code
  - word_puzzle_deque.h, word_puzzle_deque.cpp
  - Pay attention to the use of I/O streams, stringstreams, strings, and deque
  - To compile: make