CSCI 3110 Homework #5
(Barbosa)

(Points: 100)
Due: Complete and answer questions on D2L by the deadline posted for this assignment there.

Objectives: Gain familiarity with the STL and its performance by creating containers and adding and removing items from them; compare performance of the various containers.

Refer to **Stacks and Queues** slides for declarations required to implement the data structures in *a*. through *i*. below.

This assignment must be run on the ***ranger*** system, using Linux.
******* Requirement: You must answer questions for this homework assignment on D2L *******

You will have to write code to answer the questions below. **DO NOT** turn in your printed code. Simply answer the questions on D2L. The sample code below shows how you **must** time the operations on the data structures.

```
#include <time.h>
clock_t timerStart, timerEnd;

timerStart = clock();
// place code you want to time here
timerEnd = clock();
// Elapsed time in seconds is (timerEnd – timerStart)*1.0/CLOCKS_PER_SEC;
```

Express all answers as decimal seconds (e.g., 0.124 secs).

Note: Due to variability, you must run each timing at least 5 times and take the average as the answer.

Using Linux on ***ranger***, declare the following STL containers (refer to slides on Stacks and Queues on course calendar):

a. a vector that holds ints – add items using ***push_back()***
b. a list that holds ints – add items using ***push_front()***
c. a forward_list that holds ints – add items using ***push_front()***
d. a stack (with the default deque implementation) that holds ints – use ***push()*** and ***pop()***
e. a stack (implemented as a vector) that holds ints – use ***push()*** and ***pop()***
f. a stack (implemented as a list) that holds ints – use ***push()*** and ***pop()***
g. a queue (with the default queue implementation) that holds ints – use ***push()*** and ***pop()***
h. a queue (implemented as a deque) that holds ints – use ***push_back()*** and ***pop_front()***
i. a queue (implemented as a list) that holds ints – use ***push()*** and ***pop()***

******* ANSWER QUESTIONS ON D2L *******

Note: In order to conserve system memory and not negatively affect timing, make each comparison separately (e.g., do a. above, then b., then c., etc. separately)

Grading
This homework assignment will be assigned one of five categorical grades, translating to the scores listed below:
Nearly perfect (**100%**) – With minor exceptions, answer are correct or consistent with system performance
Mostly correct (**75%**) – Most of the answers are correct or consistent with current system performance
Half right (**50%**) – Only about half of the answers are correct or consistent with current system performance
Needs work (**25%**) – The majority of answers are incorrect or inconsistent with current system performance
Little to no effort (**0%**) – The work submitted is incorrect, incomplete, or demonstrates a significant lack of effort.