

CS162 Lab 9

Goals

- Implement linear data structures using STL Containers

In this lab, you will create two data structures: a stack and a queue. You will use STL containers to demonstrate basic ADTs.

Queue

For the queue, you will simulate a buffer. Remember it is first-in-first-out. You need one function that randomly generates a number. You will have a user specified percentage, and the function uses this percentage to randomly put the generated number in the buffer. For example, if the user enters 25, then there is a 25% chance of a created value been put into the queue. You will have another user specified percentage, which is the chance of a value being removed from the buffer.

The user will also enter a number for the number of turns to run your simulation. You need to record the length of the queue in each turn. After each round of the simulation, calculate and display the average length of the queue. Does your buffer behave as expected, i.e. get longer if the input chance is greater, and get shorter if the removing chance is greater?

Your program needs to handle the situation when the queue is empty and still try to remove a value. You can start the simulation with a queue of certain length, or you can start with an empty queue.

Stack

Use a stack to create a function that creates a **palindrome**, i.e. a string that is the same forwards and backwards. It does not need to be an actual word. The function will receive a string and it will return the string with the palindrome. For example, if you enter a string “abc”, your program will return “abccba”.

Testing program

Create a program for the user to test your buffer or to create a palindrome. For the queue, prompt them to enter the chance of an item arriving, the chance of an item being removed and the total number of rounds (turns). Display the results to the screen. For the stack, prompt them to enter a string. Create the palindrome and then display it.

What to submit

You will submit the following files in a zip file to TEACH:

- Code to implement your stack, both header and source files

- Code to implement your queue, both header and source files
- Code to demonstrate the operation of your stack and queue.
- A makefile.

Grading

- Programming style: 1 point
- Code to implement your stack
 - header file: 1 point
 - source file: 2 points
- Code to implement your queue
 - header file: 1 point
 - source file: 2 points
- Code to demonstrate the operation of your buffer and palindrome creator: 3 points
- No memory leak