

COSC 2436: Priority Queue

1. Introduction

Create a C++ program using your own priority queue implementation check to see if a hypothetical door is open or closed after a sequence of events.

2. Input Files

- Each valid input line will contain an instruction and a number in parentheses
- The number represents the "priority" of the instruction
 - For this assignment, a higher priority means a lower number
 - You can assume that the priorities will be unique for each instruction
- There are 4 possible instructions:
 - open the door
 - close the door
 - unlock the door
 - lock the door
- Beware of invalid inputs to ignore them
 - Example: (2)
 - Example: close the door
- There will be no extra lines and whitespaces in the input.
- Remove all `\n` and `\r` before processing any input lines.

3. Rules

- A door can only be opened or closed when unlocked.
 - Example 1: Door(Open, Locked), Instruction("close the door")
 - Result 1: Door(Open, Locked)
 - Reason: You cannot close the door while the door is locked
 - Example 2: Door(Open, Unlocked), Instruction("close the door")
 - Result: Door(Closed, Unlocked)
 - Reason: You can close the door while the door is unlocked.
- You can unlock or lock the door regardless of whether the door is closed or open.
- An instruction that has no effect on the state of the door can be ignored.
 - Example: Door(Open, Unlocked), Instruction("open the door"), the door is already opened

4. Steps

- **YOU MAY ASSUME THE DOOR STARTS CLOSED AND UNLOCKED**
- Read from the input and place each instruction into the priority queue.
- After all lines are read, process each instruction in the queue.
- Output the state of the hypothetical door.

5. Output Files

- For each input file, you will either output "the door is open" or "the door is closed"

6. Submission

- Turn in your lab assignment to our Linux server with your login credentials
- Process of Testing and Submitting
 1. In your root folder, make a folder for the assignment (lowercase with no spaces!) (Examples: lab1, hw1, ga1)
 2. Copy your program files (no spaces!) into the folder
 3. Copy your ans/input/command.txt, ArgumentManager.h, and test.sh files into the folder (NO output.txt files)
 4. In your terminal, go to your assignment folder by running **cd foldername** and run the following commands:
 - a. `chmod u+x test.sh`
 - b. `sh test.sh`
 5. Verify all cases pass and then DELETE all .txt files
 6. In your terminal, go to your root folder by running **cd** and then run the following command:
 - a. `chmod -R 755 assignment#/` (ex: `chmod -R 755 lab1/`)
 7. Log out and close

Please reach out to the TAs via email or teams for any clarifications or typos.

EXAMPLE TEST CASE

Input File

(4)
close the door (2)
open the door (4)
unlock the door (5)
open the door
close the door (13)
lock the door (9)
lock the door (3)
open the door (1)
unlock the door (6)
open the door (8)
close the door (7)

← invalid input

← invalid input

Output File

the door is open

PROCESSING PRIORITY QUEUE

open the door (1)
close the door (2)
lock the door (3)
open the door (4)
unlock the door (5)
unlock the door (6)
close the door (7)
open the door (8)
lock the door (9)
close the door (13)

Door is now open
Door is now closed
Door is now closed and locked
Door is still closed and locked
Door is still closed but now unlocked
Door is already unlocked
Door is already closed
Door is now open
Door is now open and locked
Door is still open and locked

the door is open