

Due Wed Sep 5 at the start of your lab section; Submit Server: class = cse2010, assignment = hw1SxIndividual

Due Wed Sep 5 at the end of your lab section; Submit Server: class = cse2010, assignment = hw1SxGroupHelp
 x is 2, 3, or 4—your section number, or “j” if you use java.

An online retailer provides a service that allows chatting on their website with a customer service representative. When a customer requests a chat and the retailer does not have an available representative, the customer can be put on hold. The customer decides to be put on hold or try again later. A customer can also quit waiting after being put on hold. Customers are served in the order of their requests. Similarly, representatives are selected to serve customers based on the order of when they become available.

For this assignment, assume the number of representatives is 5 and they are initially available in this order: Alice, Bob, Carol, David, Emily. To separately manage the **available representatives**, **customers on hold**, and **chat sessions**, use separate singly linked lists [in general, the number of customers requesting chats and the number of available representatives vary over time]. You may borrow programs from the textbook, but not from any other sources. We will be evaluating your submission on code01.fit.edu; we recommend you to ensure that your submission runs on code01.fit.edu.

Input: To simulate the chat requests, an input file contains the chat requests in the order of time in the same directory as your program file called hw1.c that has the main function. Your submission takes the input file name as a command-line argument. Each line is one of the following requests:

- ChatRequest *requestTime customer chatDuration waitOrLater*
- QuitOnHold *quitOnHoldTime customer*
- PrintAvailableRepList *printTime*
- PrintMaxWaitTime *printTime*

For simplicity, time is a non-negative integer (e.g., $requestTime \geq 0$) in HHMM format (HH: 00-23, MM: 00-59). When a representative is not available, *waitOrLater* indicates if the customer is going to wait and be put on hold (“wait”) or not wait and try later (“later”). QuitOnHold indicates the customer quits waiting after being on hold for too long. Sample input files are on the course website.

Output: The program prints events to the standard output (screen). Each event is on one line and possible events are:

- ChatRequest *requestTime customer chatDuration waitOrLater*
- RepAssignment *customer rep assignmentTime*
- PutOnHold *customer putOnHoldTime*
- TryLater *customer tryLaterDecisionTime*
- QuitOnHold *quitOnHoldTime customer*
- ChatEnded *customer rep endTime*
- AvailableRepList *printTime rep1 rep2 ...*
- MaxWaitTime *printTime maxWaitTimeSoFar*

For the output, note that $endTime - assignmentTime$ for a chat should be the same as *chatDuration* for the chat. *waitTime* of a chat is $assignmentTime - requestTime$ of the chat. If multiple events occur at the same timestamp, output the events in this order: ChatEnded, ChatRequest, RepAssignment, PutOnHold, TryLater, QuitOnHold, AvailableRepList, MaxWaitTime. Sample output is on the course website.

Submission: Submit hw1.c that has the main function (and other program files if applicable). Submissions from individual students are due at the beginning of their respective lab sections via assignment *hw1SxIndividual* (see the top). During the lab session on the due date, we encourage students to bring test cases (beyond the sample input) to test and improve each other’s program in the group. Improved programs are submitted via assignment *hw1SxGroupHelp*, which is due at the end of the lab section (see the top). Your program is mainly evaluated based on *hw1SxIndividual*. Improvement on test cases will receive half credit. Specifically, $testCaseImprovement(hw1) = testCaseScore(hw1SxGroupHelp) - testCaseScore(hw1SxIndividual)$; $testCaseScore(hw1) = testCaseScore(hw1SxIndividual) + testCaseImprovement(hw1)/2$. Note the late penalty on the syllabus if you submit after the due date and time as specified at the top.