

Lab Assignment 4 : Abstract Data Types - Queue

Q1. Write a function making suitable queue ADT calls to solve each of the following problems:

- Given a string, check if it is of the form $w#w$, where w is a string with alphanumeric characters only.
- Given a string, check if it is of the form ww , where w is a string with alphanumeric characters only.
- Given a string, check if it is of the form $w#w^r$, where w is a string with alphanumeric characters only, and where w^r stands for the reverse of the string w .

Q2. Use queue to solve the following customer scheduling problem

Problem : Write two **BakeryQueue** functions that manage a waiting line of customers to be processed at a bakery shop.

Part 1 : In the **BakeryQueue_FIFO** program, you need to manage the customers who stand in a queue following the FIFO rule : each customer is assigned a token on arrival at the bakery. The person with the smallest token number stands at the head of the queue followed by the next smallest token and so on. The person standing at the head of the queue would be next person to be served by the delivery boy.

Input : `customerId`, `token` (consider an integer)

Output : Order of processing the customers following the FIFO rule.

Consider, a list (provided as a file having `customer id` and `token` in each line of the file) that can accommodate **n (say 10 initially)** customers at a time. Every new customer having a `customer id` and `token` henceforth is enqueued accordingly.

Part 2 : Round Robin .

Customers decide to buy n items in n turns. For example if customer A has to buy 10 items he/she takes 10 turns to buy them. In the first turn, he/she buys item no. 1 when his/her turn comes in the queue, and then goes back and joins the queue at the rear position to attempt buying item 2 and so on.... Write a function **BakeryQueue_ROUND_ROBIN** to implement the above scheduling policy.