[PROGRAMMING]

CASE DOCUMENT

CONFIDENTIAL UNTIL COMMENCEMENT OF COMPETITION

CASE SYNOPSIS

You are the Quality Assurance Engineer for the automated crane that is being installed at an e-commerce company. You have the task to develop a test software for the automated crane that will make sure that the orders are stacked correctly before they go out for shipment. Your test software will simulate the updating of orders and rearrange them accordingly.

GENERAL RULES

- All questions regarding the competition problem should be asked during the welcome and briefing session. Questions asked during the build phase must be done so through Slack and the response subject to be shared with all competitors.
- Each competitor is expected to have a computing device with a stable internet connection for the duration of the competition.
- Visitors and non-competing individuals are not allowed to engage in the competition. Violation of this rule will result in the entire team's disqualification.
- Teammates may not be physically present with one another in the same room. This is to ensure fairness to all participants who may not have the same access capabilities due to location and/or lockdown restrictions.
- Final design and presentation materials must be submitted to the submission form prior to the end of the design and build stage. It is the team's responsibility to ensure that they are submitted on time.

SCHEDULE

The time slot for each team's presentation will be random and will be given at the conclusion of Saturday's work sessions. Each group is expected to be prepared and ready to go for the first slot and is expected to remain as such until after their presentation. The timeline for the Programming Comp is found at the end of this document.

The final submission is due at 5:20 pm PST. To account for potential issues, submissions will be accepted until 5:30 pm PST. After 5:30 pm, a 2% penalty will be applied for every minute your team goes over time up to a maximum of 40%, after which submissions will not be accepted.

BACKGROUND

QuickBuy is a successful e-commerce website that receives over thousands of orders everyday. QuickBuy's CEO is very ambitious and wishes to reduce half its warehouse manpower by introducing automated technology for the warehouse. One of the latest inclusion is an automated crane that can arrange orders according to the delivery location. You are the Quality Assurance Engineer for the automated crane and will be working to make sure the crane is installed correctly and works as desired. During your testing period at the company you realize that when customers update their orders the arrangement of orders is changed. You discuss this with your team and decide to develop a test software that makes sure that the stacks of orders are arranged correctly.

The test software functions in the following way:
- Accepts the input configuration
- Accepts the process configuration
- Produces an output configuration
- If the output procedure is rearranged due to updated orders, it generates a new process configuration
- Finally, it produces the desired configuration and arranges the stacks of boxes accordingly

RESTRICTIONS/SPECIFICATIONS

The automated crane takes the following inputs and performs the corresponding actions:
- 1: Move left (moves to the next stack on the left. Does not do anything if already in the leftmost position)
- 2 : Move right (moves the next stack on the right. Does not do anything if already in the rightmost position)
- 3 : Lift a box from the stack (Does not do anything if the crane already has a box)
- 4 : Drop a box on the stack (Does not do anything if the crane doesn't already have a box)
- 0 : Quit

Further, there is a limit *4* on the number of boxes on each stack. If a 'drop' command would result in a stack having more than *4* boxes, the crane ignores this drop command. If the current stack has no boxes, a 'pick up' command is ignored.

Starting point for the crane will be the left-most stack.

Use the example below to understand the different configurations required to run the automated crane
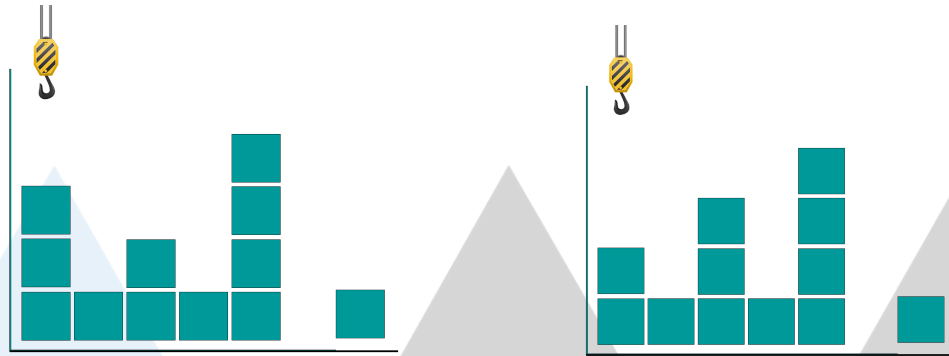
First image is the initial stack and the second image is the output stack arrangement.

According to this examples
Input configuration : 3 1 2 1 4 0 1
Process configuration : 3 2 2 2 2 4 1 3 1 4 0
Output configuration : 2 1 3 1 4 0 1

OBJECTIVE

This problem is broken into 3 levels. Complete as many as you can, however they must be completed in order.

Level 1A : Backend (Input to Output)
Create a program that converts the input configuration to output configuration using the process configuration according to the specifications/restrictions of the crane mentioned above.

Sample input:
3 1 2 1 4 0 1
3 2 2 2 2 4 1 3 1 4 0
Sample output:
2 1 3 1 4 0 1

Level 1B : Backend (process configuration)

Create a program that uses the output and input configuration to find out the process configuration.

Sample input:
3 1 2 1 4 0 1
2 1 3 1 4 0 1
Sample output:
3 2 2 4

Level 1C : Backend (rearranged stack)
Create a program that rearranges the output stack randomly.

Sample input:
3 1 2 1 4 0 1
Sample output:
3 1 1 2 4 0 1

Level 2 : Create UI and connect to Backend
Create an user interface that allows a worker to enter the input configuration and the process configuration to obtain the output configuration. Create the GUI to show the stacks of boxes.

By the end of Level 2 your program should follow this flow of execution:
1. User enters the input and process configuration to obtain the output using the backend made in Level 1A.
2. Rearrange the stacks according to the backend algorithm made in Level 1C.
3. Convert the rearranged stack to the final stack using the backend algorithm made in Level 1B.

Level 3 : Advanced features
Let the user rearrange the stack instead of randomly doing it in the backend. This will allow you to simulate the rearranging of stacks due to updating the orders. You can also use any other way of representing the input, output and process configurations, without interfering with the general flow of automated crane.

DELIVERABLES
You must write and submit a design document. This document should describe your process through the competition (how did you divide work, why did you choose the platform you did, thought process for different data structures and algorithms, etc). This is due at the same time as your code, and will be submitted together. You must additionally submit user documentation, this can be as simplistic as a

github README.md or a full separate user manual but it must sufficiently document the program and how to build/use it.

Make sure you submit all files needed to execute both the front end (client) and back-end (server) along with clean and maintainable source code (comments, good separation of presentation layer).

JUDGING RUBRIC

The following rubric is meant to guide judges' decisions and evaluation of competitor performance. All questions/descriptions are to be used with judges' discretion. Scores will be determined based on the judges' final evaluation of the competitor(s).

Design

Program Design

Are the design choices and analyses justified? Are supporting arguments provided (where applicable)? Does the design follow accepted industry best practices?

Solution Performance

Does the solution perform well? Is it likely to be scalable? Does it solve the problem as presented?

Innovation and Creativity

How creative is the solution? Does it apply new technology or solutions or old technology in novel ways?

Appropriate User Documents

Does the solution contain appropriate user documents as applicable? User manuals and design documents.

Presentation

Presentation/Organization

Is the subject matter communicated clearly? Does the speaker provide necessary background information prior to addressing specific topics? Does the flow of the presentation contribute to audience understanding?

Speaker Presence/Confidence

Does the speaker demonstrate a high level of understanding regarding the subject matter? Is the speaker engaging? Does the speaker demonstrate strong communication skills and an understanding of how to engage and educate audiences?

Does the speaker respond confidently to questions? Does the speaker demonstrate an understanding of the topic beyond the presentation material?