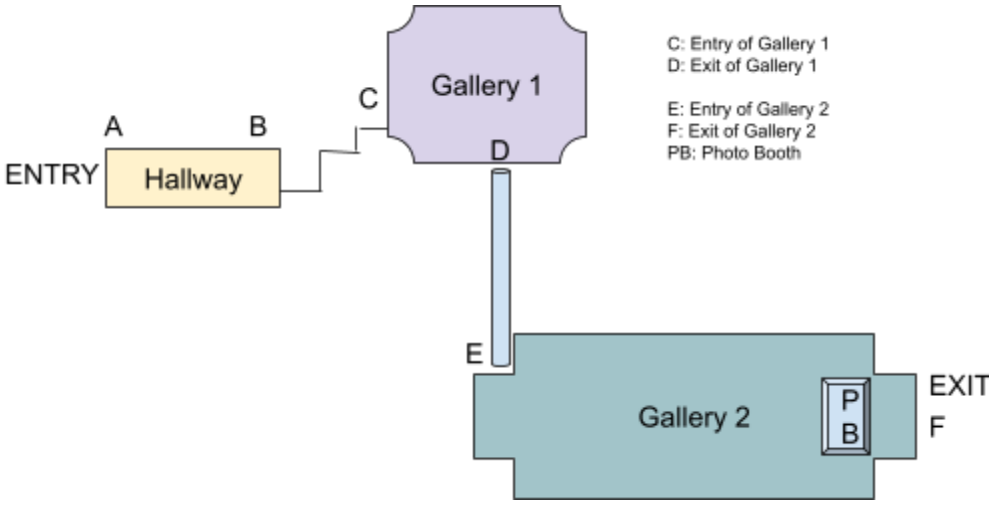
Bangladesh University of Engineering and Technology
CSE 314: Operating Systems Sessional
IPC Assignment

A Visit to a Museum

A group of **N** friends went to visit a museum. There are two categories of entry tickets: premium and standard. Each member of the mentioned group purchased a standard ticket. At the time of their visit, there are **M** other visitors. Let's assume that every other visitor purchased a premium ticket.

At the entrance, each visitor is handed over a map of the museum. There is only one entry gate and one exit gate of the museum. The map looks like below.



Any number of visitors can enter the Museum through the ENTRY point. Each visitor spends **w** time units of time in the hallway AB to see the arts displayed on the hallway. A visitor then moves towards Gallery 1. A visitor spends **x** units of time in Gallery 1. The maximum occupancy of Gallery 1 is 5. One can reach Gallery 1 from the hallway through stairs of 3 steps. You may add small delays at each step. On every step, there can be at most one person moving in an upward direction. No overtaking or skipping a step is allowed here. One can enter through C only if there is room for a visitor inside Gallery 1.

Task 1: Your task is to implement the moves of a visitor using synchronization and locking mechanisms. Print every move of each visitor (with timing info).

From Gallery 1, the next spot to visit is Gallery 2. The two galleries are connected through a long narrow glass corridor DE. Only 3 visitors can walk simultaneously through the glass corridor DE. Add some arbitrary delay to walk through the glass corridor for each visitor. The

maximum occupancy of Gallery 2 is as large as it can fit all visitors. In other words, if any visitor can cross the channel DE, he can enter Gallery 2.

Task 2: Your task is to implement the moves of a visitor using the producer-consumer mechanisms making sure that every visitor that enters the museum enters Gallery 2. Print every move of each visitor (with timing info).

Inside Gallery 2, there is a photo booth which requires scanning of the entry ticket again. A visitor spends **y** units of time in Gallery 2 before reaching the photo booth. Every premium ticket holder gets an *immediate* and *exclusive* access to the booth. Standard ticket holders get *shared* access to the booth. There is a waiting area in front of the photo booth. If there is a premium ticket holder and a standard ticket holder in the waiting area, the premium ticket holder gets priority over the standard ticket holder. [The scenario sounds familiar, right!!] A visitor spends **z** units of time inside the photo booth.

Task 3: Your task is to implement this part using the logical construct of the reader-writer problem (with timing info). You are allowed to add some random intervals with the time unit **y** (visiting time in Gallery 2) to generate the following scenarios. i) When a standard ticket holder is waiting in the waiting area, a premium ticket holder arrives and gets priority over the standard ticket holder, ii) When a standard ticket holder is inside the photo booth, another standard ticket holder arrives and gets shared access to the booth.

1. Implementation Guideline

- There **MUST NOT** be any busy waiting anywhere in the implementation.
- You can generate all visitors at once and assign each visitor an unique ID. Choose an ID for a standard ticket holder between 1001-1100 and an ID for a premium ticket holder between 2001-2100. However, to achieve randomness, you should add random delays before a visitor steps into the hallway AB.
- Use Poisson distribution to generate random numbers. Use a suitable inter-arrival rate.
- The lifetime of a visitor inside the museum is from ENTRY = A to EXIT = F. If you decide to complete only Task 1, make EXIT = C and change the maximum occupancy of Gallery 1 to a large number. If you decide to complete up to Task 2, make EXIT = E and do not change the maximum occupancy size of Gallery 1. Here, ENTRY corresponds to starting a visitor thread and EXIT corresponds to finishing/killing that thread.
- The timing of the operations should be implemented using sleep. The relative time for each operation is given as input.
- Input/Output:
    - You will take the following values as input from the command line
        - N M

w x y z

Where N = Number of friends visiting the museum/ standard ticket holders
M = Number of other visitors /premium ticket holders
w, x, y, z = relative time units for the operations

2. Sample Input/Output

Up to Task 1:

Input	**Output**
10 0 **2 0 0 0**	Visitor 1001 has arrived at A at timestamp 3 Visitor 1004 has arrived at A at timestamp 4 … Visitor 1001 has arrived at B at timestamp 5 Visitor 1004 has arrived at B at timestamp 6 … Visitor 1001 is at step 1 at timestamp 6 Visitor 1001 is at step 2 at timestamp 7 Visitor 1004 is at step 1 at timestamp 7 Visitor 1001 is at step 3 at timestamp 8 Visitor 1004 is at step 2 at timestamp 8 … Visitor 1001 is at C (entered Gallery 1) at time 9 Note: The timestamps and the relative order of the print statements are imaginary here. In your program you will calculate the times based on the interaction between the system and the visitor. <u>You only need to make sure that the timestamps of a visitor increases with step count. Also, if timestamp of visitor 1001 in step 1 is smaller than timestamp of visitor 1004 in step 1, then the timestamp of visitor 1001 in step 2 is smaller than timestamp of visitor 1004 in step 2.</u>

Up to Task 2:

Input	**Output**
10 0 **2 6 0 0**	(similar output as above) Visitor 1001 is at D (exiting Gallery 1) at time 15 … Visitor 1001 is at E (entered Gallery 2) at time 17 Visitor 1004 is at E (entered Gallery 2) at time 18 Visitor 1002 is at E (entered Gallery 2) at time 19 …

	You need to make sure all visitors enters Gallery 2.

Up to Task 3:

Input	**Output**
10 6 **2 6 6 3**	Visitor 1001 has arrived at A at timestamp 3 Visitor 2004 has arrived at A at timestamp 4 … Visitor 1001 is about to enter the photo booth at timestamp 24 Visitor 1004 is about to enter the photo booth at timestamp 25 Visitor 2004 is about to enter the photo booth at timestamp 26 … Visitor 1001 is inside the photo booth at timestamp 25 Visitor 2004 is inside the photo booth at timestamp 28 Visitor 1004 is inside the photo booth at timestamp 31 In this output, 2004 gets priority over 1004.

3. Marks Distribution

Tasks	Subtask	Marks
Task 1	Random arrival of visitors	**5**
	Implementation of steps using locks	**20**
Task 2	Implementation of producer-consumer for Gallery 1 occupancy limit	**15**
	Implementation of producer-consumer for Glass Corridor	**10**
Task 3	Implementation of reader-writer problem	**30**
Printing with timing information		**10**
Presenting your output during evaluation showing different scenarios effectively		**10**

4. Submission

- Create a directory by your 7 digit student name (2005XXX).

- Put your source code inside.
- Zip the folder, rename it to 2005XXX.zip.
- Submit the zipped folder.

Deadline: November 16, 2024. 11:55 PM.

Plagiarism Policy

- –100% marks will be deducted for plagiarism.
- Work on the problem on your own.