

Assignment 5

Problem Statement

You have been asked by the student council to monitor the use of washing machines in OBH. You have to report the events of the day in the format provided by the council with some statistics. In your conclusion line, you have to report if more washing machines are needed or not. More washing machines are needed if at least 25% of the students who came to the machines returned without getting their clothes washed.

During the entire day:

- N students will come to get their clothes washed
- There are M functioning washing machines
- The time at which the i-th student comes is T_i seconds after the execution of the program
- The time taken to wash the i-th student's clothes is W_i seconds
- The patience of the i-th student is P_i seconds, after which he leaves without getting his clothes washed

Statistics:

- Total number of students who left without washing. Integer between 0 and N.

Given appropriate inputs, simulate the events using multi-threading. Avoid deadlocks and busy waiting. Instead, use semaphores and mutex locks to implement the problem. You are required to write your algorithm, followed by implementational details in your report.

Input format:

- The first line contains 2 integers N and M.
- The next N lines contain 3 integers T_i, W_i and P_i, the i-th line corresponding to the detail's of the i-th student

Output format

1. A relevant line has to be printed in the given color for the following events:
 - A student arrives to get their clothes washed (White)
 - A student gets an empty washing machine to wash his clothes (Green)
 - A student leaves after his washing is complete (Yellow)
 - A student leaves without getting his clothes washed (Red)
2. A single integer stating the number of students who came but could not wash their clothes.
3. A final line saying "Yes" if more washing machines are needed, or "No" if they are not needed. (in White)

Sample Test Cases

Test Case 1

Input:

```
5 2
6 3 5
3 4 3
6 5 2
2 9 6
8 5 2
```

Expected output:

Student 4 arrives
Student 4 starts washing
Student 2 arrives
Student 2 starts washing
Student 1 arrives
Student 3 arrives
Student 2 leaves after washing
Student 1 starts washing
Student 5 arrives
Student 3 leaves without washing
Student 1 leaves after washing
Student 5 starts washing
Student 4 leaves after washing
Student 5 leaves after washing
1
No

Test case 2

Input:

3 1
2 5 1
1 2 4
2 4 2

Expected output:

Student 2 arrives
Student 2 starts washing
Student 1 arrives
Student 3 arrives

```
Student 2 leaves after washing
Student 1 starts washing
Student 3 leaves without washing
Student 1 leaves after washing
1
Yes
```

Notes

- Assignment must be implemented in C or C++.
- To connect to your ada account:

```
ssh <username>@abacus.iiit.ac.in
```

- You will be prompted to enter your password, post which you will gain access to the home directory to the ada instance. Create a new file and develop your script there. To copy the script back to your local machine, use secure copy:

```
scp <username>@abacus.iiit.ac.in:/path/to/solution.c /path/to/lo
```

- If there are multiple students waiting, the student who arrived first will use the machine first (FCFS)
- If there are multiple students arriving at the same time, the one with the lower index will go first
- If a student arrives at time t_i and has patience p_i , he can use the machine at time $(t_i + p_i)$ if it gets empty at that second
- If a student arrives at time t_i and has patience p_i , he will leave at time $(t_i + p_i)$ if there are no machines empty at that second
- Since the output of multi-threaded semaphores can be non-deterministic, you will not be penalised if FCFS is not followed on some runs. The order of the statements printed above can vary depending on implementation, hence

different runs of the same input might have varying orders for printing, but the content of the statements must not change. Even though the final output may vary, implementing FCFS is mandatory, and it should be done by the threads.
Using a queue would lead to negative marking.

- It is recommended to use a mutex for printing to avoid interleaved statement printing.
- Follow the exact format mentioned in the test cases above.
- Include instructions for compilation and assumptions in your readme.

Submission Format

```
<roll_no>_assignment_5  
  | -> readme.md  
  | -> solution.c
```