Note:

- *Submission Deadlines (physical submission need to be done personally in my office):*
  - **Submit Q1 [GCR snapshots] on or before 27th Feb 2024 11:59 PM and get 0.5 bonus weightage.**
  - **Q1 and Q2 [physical + GCR] -> 5th March 2024 11:59 PM**
- *This whole assignment should be done by hand. Use pencil only.*
- *Grading will be done for dry run only in Q1. Copying code is only to create context.*
- *Cheating and plagiarism will result in a deduction of weightage up to 20.*
- *You are allowed to brainstorm and discuss Questions # 2. However, write your own code on paper without help.*
- *Both questions have equal weightage.*
- *I reserve the right to take viva and/or validation quiz to validate learning outcomes of this assignment.*

Questions 1:

a) Copy the code given in the following figures in chapter # 3 on paper. **Do an in-depth dry run:** [60%]

i. Linux process control block structure (no dry run. Rather explain what goes in each member of this structure).
ii. Figure 3.8
iii. Figure 3.12 and 3.13 (producer consumer)
iv. Figure 3.16 and 3.17 (shared memory)
v. Figure 3.21 and 3.22 (anonymous pipes)
vi. Named pipe code given in chapter # 3 lecture slides.

b) **Show dry run to work-out the outputs** on the following code snippets from practice problems and exercises sections of chapter # 3. [40%]

Practice Problems:
i. Figure 3.30, 3.31
Exercises (find figure number in the right sections)
ii. Figure 3.21, 3.22, 3.23, 3.24

Question 2:

a) Consider the producer-consumer code of Figures 3.12 and 3.13 including its explanation on pages 126-127 and in Figures 3.16 and 3.17 to create and access a shared memory area named “CIRCULARBUF”. Now, write a complete producer-consumer program with one producer and two consumers consA () and consB (). The producer generates random message of size 120 characters by calling a function gen_msg (). ConsA () prints messages starting with tag MSGA and messages with tag MSGB will be printed by consB (). [60%]

b) Now suppose that we want to relax the tag restriction on consumer processes and allow them to check shared memory in turn and print the message if it arrives when they check them.

Given detail C code or pseudo code examples how you change portions of your code in part (a) above to implement this requirement. Your approach should be based on the syllabus covered at the time of releasing this assignment and could have problem which we haven’t covered yet. [40%]

----------(X)----------