EE 337: Microprocessors Laboratory (Spring 2025)
Electrical Engineering
Indian Institute of Technology Bombay
ENDSEM: 40 points (Duration: 3 hours) Date: April 12, 2025

Instructions:

- You are allowed to use **only your** codes from previous labs. Sharing material or old codes is strictly not permitted.
- Use of internet during the course of this examination will be considered as copying and strict action will be taken.
  - **You must put your laptops in airplane mode.**
  - **No internet browser should be opened during the exam**
  - **Use of LLMs and any kind of AI assistance (online or offline) is strictly prohibited, if found using treated as malpractice and appropriate action will be taken.**
  - **Install a non-browser pdf reader to read pdf files during the exam and set that as default pdf reader.**
    Examples are Acrobat Reader, Nitro PDF Reader, Sumatra PDF Reader.
- Any discussion during exam is not permitted.
- **Optimize your code and use short print statements** in UART serial terminal as it can make your memory out of bound.
- **Only use header files provided, don't use any other header file** as it may lead to memory go out of bound.

Design Flow

You will implement a menu-based ticket booking system simulating UART communication at **baud rate 1200** for user interactions.

Q1: User Authentication (10 points)

Implement a secure user login system as follows:

1. Prompt the user to enter **Login ID** and **Password**. Verify these inputs against predefined credentials stored in memory (variables). It should check with predefined credentials only after taking both **Login ID** and **Password**. If any of the inputs doesn't match then display **"Invalid"** and start again. Only **one user** credentials should be stored.
   Store the following credential:
   Login ID will be your name (maximum first 6 characters if name is longer than 6 characters) and password will be first four letters of your name and last 4 digits of your roll number respectively.
   For example, Name: John Doe, Roll Number :23B5525.
   Login ID: john
   password: john5525
2. Implement a security verification step by generating a random **3-digit Captcha** number. Prompt the user to input this Captcha correctly before proceeding. Captcha will be displayed on the serial terminal and you will input this captcha through your serial terminal to verify. If wrong input entered, then display **"Invalid"**, regenerate another captcha value using the updated seed value and prompt the user again for captcha.

Algorithm to generate 3 digit random number:
seed = ((seed * c1) + c2) % 1000
where c1 and c2 are constants
seed = last 4 digits of your roll number
c1 = 1204
c2 = 2025

3. After successful authentication, display the **user wallet balance** (use initial wallet balance = Rs. 5000) on the serial terminal and if you are displaying the wallet balance after a successful booking, updated wallet balance should be displayed.

Store login credentials, wallet balance, and captcha temporarily in C variables of your choice.

Q2: Ticket Booking Procedure (5 points)

After successful login, implement the ticket booking procedure:

1. Show a list of predefined train services on the serial terminal, with costs specified per person.: Format: <Train No.><Train Name><Amount>
   - 1 Rajdhani Express - Rs. 1500
   - 2 Chennai Express - Rs. 1000
   - 3 Hapa Duronto - Rs. 1200

   Allow the user to select Train No. of one train via UART.
   For any invalid input, raise an error **"Invalid Train No."** on serial terminal and prompt the user again from **Ticket Booking procedure**.
2. Take the number of passengers as input (maximum 4) via UART.
   For any other invalid input, raise an error **"Max. 4 passengers allowed"** on serial terminal and prompt the user again for **number of passengers**.
3. Compute the total cost:

   $$\text{Total Cost} = \text{Number of Passengers} \times \text{Ticket Price per Passenger}$$

   Display total cost on the serial terminal.

Q3: Payment and Booking Confirmation (10 points)

1. Prompt the user for confirmation of the transaction like display **"Confirm Txn (y/n):"** and ask for a user input.
2. Upon confirmation(y), compare wallet amount with calculated ticket cost, else(n) display **"Txn failed"** and restart the **ticket booking procedure**.
3. If funds are insufficient, terminate and display **"Txn failed: Insufficient Balance"** on serial terminal and restart from the **ticket booking procedure**.
4. If funds are sufficient, deduct cost from wallet balance and update variable. Display the message **"Booking Successful"**, show the updated wallet balance on serial terminal and restart from the **User Authentication**.

Ensure that all user interactions and calculations are handled effectively through UART modules. After completion of question-3, save the main.c code in a textfile so that you won't lose any code while doing the question-4.

Q4: Timer-based timeout for security (15 points)

1. Implement a timer-based (**10 seconds**) timeout for security between **Payment confirmation and amount deduction from wallet**.
2. Display the OTP (generate using the same random number generator function used in first part starting with updated seed value) on serial terminal. Take the OTP as input on the serial terminal for verification.
3. If OTP is not entered in the given time, raise an error, i.e., display **"Txn failed: Timeout"** on serial terminal and restart from the **payment and booking confirmation** of transaction.
4. If wrong OTP entered, raise an error, i.e.,display **"Txn failed: Invalid OTP"** on serial terminal and restart from the **payment and booking confirmation** of transaction (will be evaluted only if previous step works).
5. If correct OTP is entered in the given time, then continue from **amount comparison and deduction from wallet- Q3.2** (will be evaluted only if previous step works).

Rubrics

- **Q1:** User Authentication: **10 points**
- **Q2:** Ticket Booking Procedure: **5 points**
- **Q3:** Payment and Booking Confirmation: **10 points**
- **Q4:** Timer-based timeout for security: **15 points**