EE 337: Microprocessors Laboratory (Spring 2022)
Indian Institute of Technology Bombay
ENDSEM: 35 points (Duration: 3 hours) Date: April 02, 2022

Instructions:

- You are **ONLY** allowed to use your codes from previous labs and the files included in the **endsem-references.zip** posted on Moodle. Sharing material or old codes is 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**
  - **Install a non-browser pdf reader to read pdf files during the exam.** Examples are Acrobat Reader, Nitro PDF Reader, Sumatra PDF Reader.
- Any discussion during exam is not permitted.
- For Question 2, it is advised to add features one by one as mentioned in the rubrics section to maximise your chances of getting partial credits if you are not able to finish within time
- At the completion of **every question**, **show your solution to an invigilator TA** for evaluation.

1. [10 points] SIMULATION BASED QUESTION

   Matrix Addition using assembly language programming.

   You must use assembly language to answer this question. The goal here is to add two $N \times N$ matrices M_1 and M_2 having 1 byte entries, and store their sum $M_1 + M_2$. **You can assume that the entries in the input matrices M_1 and M_2 are in the range 0 to 127.**

   The value of N can vary from 2 to 4. At memory location 30H, the value N will be stored. This indicates the size of the matrix.

   The two matrices and their sum must be stored into contiguous memory locations starting at 31H. For instance, if the value stored at 30H is 3, then the 9 values for M_1 would be stored from 31H to 39H and for M_2 the values will be stored from 3AH to 42H. In this implementation, result must be stored starting from address 43H to 51H.

   A hard-coded implementation where all N^2 additions are performed using separate instructions will **fetch only 5 points** out of the 10 points. For example, the following code is a hard-coded implementation:

```
if size == 2,
  (39H) = (31H) + (35H);
  (40H) = (32H) + (36H);
  ...
if size == 3,
  (43H) = (31H) + (3AH);
  ...
if size == 4,
  (80H) = (31H) + (40H);
  ...
```

   Using one or more loops will get you **full credit**. If not for memory costraints, your code should work for $N = 5, 6$ also just by changing the byte stored at location 30H.

2. [20 points] IMPLEMENTATION BASED QUESTION

   In this question, you will be writing a program to simulate the behaviour of an ATM which can dispense notes having denominations 500 and 100. The actions of the ATM user will be simulated using key presses on a keyboard connected to Pt-51 using UART with baud rate 9600.

   The denominations available are of Rupees 500 and 100 respectively. Assume there is an infinite pool of notes available. There are 2 accounts available with details as follows

   - Account No.: 1; Name: Sita; Password: EE337
   - Account No.: 2; Name: Gita; Password: UPLAB

   Assume both accounts have Rupees 10,000 available initially. You need to take care of the following cases. There are **three** states of operation.

   1. **Initial state**
      - Upon reset, the kit should enter the initial state.
      - Whenever the kit is in the initial state, the UART display must show the following message:
        > "Press A for Account display and W for withdrawing cash"

        HINT: Use "\n" for encoding a newline character at the end of your display messages. For Windows, use "\r\n".
   2. **State A: Account display**
      - This state is for displaying account balances. It is entered when the user types the character "A" or "a" through UART. When this happens, UART display must show the following message
        > "Hello, Please enter Account Number"
      - When a **valid** account number is entered, the following information should be displayed on UART.
        > "Account Holder: <insert name>"
        > "Account Balance: <insert figure>"
      - For instance, if the user typed 1, then the information displayed would be
        > "Account Holder: Sita"
        > "Account Balance: 10000"
      - For an **invalid** account number, the following must be displayed
        > "No such account, please enter valid details"
      - Irrespective of whether the account number is valid or invalid, the kit should **go back to the initial state.**
   3. **State W: Withdraw cash**
      - This state is for withdrawing cash. This state is entered by pressing "W" or "w" in the initial state. After this state is entered, UART window should display
        > "Withdraw state, enter account number"

- The response to an account number entry must be same as described in state A. That is, for a valid account number the account holder and balance are displayed. For an invalid account number, the error message is displayed.

- For a valid account, the following must be displayed after displaying balance:
  > "Enter Amount, in hundreds"

- Here, you must input exactly 2 digits. The requested amount would be that number multiplied by 100. For instance, if I enter the number 42, the amount I want is Rupees 4,200. If I enter 07, the amount needed is Rupees 700 and so on.

- If the entered character is not a number, then display
  > "Invalid Amount"
  In this case, the kit should **go back to the initial state.**
- For a valid number, check whether the account has sufficient balance. If not, then display
  > "Insufficient Funds"
  Even in this case, the kit should **go back to the initial state.**
- If a valid number is entered for a valid account with sufficient balance, then

  – The **MINIMUM** number of notes required to pay the requested amount should be calculated.
  – The minimum number of notes should be displayed on UART (as described below)
  – The amount must be deducted from the account.
  – Details of the number of 500 and 100 Rupee notes dispensed and the updated account balance should be displayed on the UART display as follows.
  > "Remaining Balance: ZZZZ"
  > "500 Notes: XX, 100 Notes: YY"

RUBRICS:
Please show state-wise progress to your TA. This is to help us give you partial marks. You might have some state working and later your code may not work due to the changes you have made. In that case, you can get partial credit if you show the previous state working to a TA.

The following partial points would be awarded.

- Initial account balance check state running correctly: 4 points
- Able to take one digit input for amount and dispense it without deducting balance: 6 points
- Able to reduce account balances after transactions: 5 points
- Able to take 2 digit inputs: 5 points
- **Note:** 3 points will be deducted for an implementation dispensing non-minimum number of notes

3. [5 points] **Make a copy of your question 2 solution before trying this question.** Integrate the password feature into the ATM designed in question 2. As soon as a valid account number is asked, the password must be asked by showing the following on the UART display:

> “Please enter password”

Only if correct password is entered, the amount entered must be dispensed