1. [5 points] In this experiment, you will learn to display content on the LCD connected to the Pt-51 kit. Download the `lcd.asm` file and `lcd-control-made-easy.pdf` from Moodle. The latter has general information about LCD operation which is helpful in understanding the code in `lcd.asm`.

   - Compile `lcd.asm` and load the hex file on to the kit. Make sure the output on the LCD screen is as shown below:

     ```
         Pt-51
     IIT Bombay
     ```

   - Study the subroutines used in the `lcd.asm` code and their functionalities. Modify `lcd.asm` to display "EE337-2022" on the first line and your **first name** on the second line (truncate to 16 characters if you have a longer name). **Pad the display lines with spaces such that these are centered on the LCD when displayed.** You should load and run this program on the Pt-51 kit.

2. [5 points] Write an assembly language subroutine which converts an 8-bit number stored at 30h to its ASCII representation which needs to be stored at 60h and 61h. Note that each nibble of the 8-bit number has an 8-bit ASCII representation. For example, if 1Fh is stored at 30h then the locations 60h and 61h should hold the values 31h and 46h. *Hint: Look at the ASCII table at* `https://commons.wikimedia.org/wiki/File:ASCII-Table-wide.svg`

   This subroutine will be used in the next question. You need to upload a separate file for this question.

3. [10 points] In lab 4, you had written values to memory using the `MOV` instruction before uploading the code. However, that cannot be changed after compilation. To change values during the running of a program, we can use the on-board DIP switches, which were used last time to choose the tests to conduct. Your program will cycle through 7 states (State 0 through State 6).

   State 0: This is a reset state. All 4 LEDs must glow for a period of 1 second, before unconditionally moving to State 1. During State 0, the following text must be visible on the two rows of the LCD:

Enter Inputs
EE337-2022

   State 0

   States 1-4: Use the DIP switches as the input mechanism and take 4 nibbles as input to make two 8 bit numbers $N1$ and $N2$.

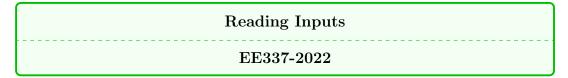   - The order in which the nibbles are to be entered is the following:
     - Higher nibble of first number ($N1_H$).
     - Lower nibble of first number ($N1_L$).
     - Higher nibble of second number ($N2_H$).

- Lower nibble of second number ($N2_L$).

- Each DIP switch represents a bit. The on position is 0 and the off position is 1.

- For example, to input a nibble 1001 the four switches need to be moved to the state off-on-on-off.

- The state of the switches should be read after every **two seconds**. This means that you have to bring the switches to the correct state before the sampling instant. The sequence of events will be the following:
  - Suppose State 1 is entered at $t = 1$ second
  - Move the nibbles to the positions corresponding to $N1_H$ before $t = 3$ seconds.
  - The state of the DIP switches is read at $t = 3 + \epsilon$ seconds.
  - Move the nibbles to the positions corresponding to $N1_L$ before $t = 5$ seconds.
  - The state of the DIP switches is read at $t = 5 + \epsilon$ seconds.
  - And so on for $N2_H$ and $N2_L$

- You can reuse the delay function used in the last lab to create the 2 second delay.

- To know which nibble is being taken as input, while taking each nibble, only one LED must be on. All other LEDs must be off. The LED to nibble mapping is as follows:

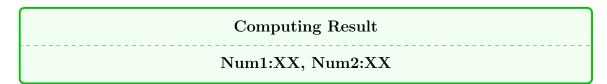| $N1_H$: P1.7	$N1_L$: P1.6
$N2_H$: P1.5	$N2_L$: P1.4

Table 1: LED to Nibble Mapping

- During the reading of all four nibbles, the following text must be displayed on the two rows of the LCD.

Reading Inputs

EE337-2022

States 1 - 4

- Store the four nibbles at memory locations 30h and 31h.

State 5: During State 5, the LEDs must be off, the numbers must be displayed on the LCD for 2 seconds, in the following format where XX is the number in hexadecimal format.

Computing Result

Num1:XX, Num2:XX

State 5

Meanwhile compute the multiplication of these two numbers using the `MUL` instruction and store the result in the memory locations 50h and 51h. Use 50h for the least significant 8 bits of the result and 51h for the most significant 8 bits of the result.

State 6:

The result of the multiplication must be displayed along with the original numbers in the following format. The system must stay in this state until reset button is pressed, which should make the system go to State 0 again.

> **Result = XXXX**
> --
> **Num1:XX, Num2:XX**

State 6

Make sure you change the default clock frequency to 24MHz (as it is on the PT-51 board). This is a fairly large code with a number of subroutines. Locations for LCD subroutines is C:200h and Strings are at C:300h. Ensure that the main function and other subroutines start at code memory 100h. If adequate space is not given between different sections of code, you may unintentionally overwrite over an address space which already had some useful code.

TA Checkpoints

1. Check the understanding of LCD operation and the `lcd.asm` code.

2. For question 1, check that the text is properly centered.

3. For question 2, check that the byte stored at 30h is represented correctly as ASCII in locations 60h and 61h.

4. For question 3, check that the results and delays are correct in Keil debugger.

5. For question 3, check that the results and delays on the board.