Introduction:
Areas covered in this Lab: Microsoft's Assembly Language Development System (MASM) and Parallel Port Exerciser (PPE). This exercise will use the Parallel Port Exerciser (PPE) board.

Parallel Port Exerciser (PPE) Board:
Check out a PPE board from your lab instructor and connect it to one of the computers in the lab.

1. Review the information on the PPE board (switch addressing, schematic, and pictorial diagrams).

2. Assemble the PPE Test program, and verify it and the PPE board are working. Verify you can select the “1”, “2”, and “3” keys (and that the program recognizes the correct key) and modify the program to have the key value displayed on the computer monitor. Also, modify the program to print a title with your name in it. Include this in your demonstration to your lab instructor. Expand the program to recognize and identify all keys (0-9, # and* keys) and switches (red and white switches) on the PPE and display the result on the computer monitor.

3. Basic Input and Output routines: Write a program that gets a number from the number pad of the PPE, output this number on the most significant nibble of the 8 LEDs, and echo this number to the screen of the computer with a prompt. Have your program continue to get numbers by staying in a loop. Terminate the program when the RED button is pressed. The program should start in the “enter a number mode” with all zeros displayed. Echo the numbers entered and the results to the monitor.

4. Create a Two Function Calculator (addition and subtraction). This will require a series of subroutines to: clear the screen, enter keyboard data into a string buffer, print the contents of a buffer to the screen, and pack (and unpack) numbers into BCD format. In addition, this will require subroutines to add and subtract numbers (in BCD format). The final program will have two modes (use a menu program); one using the computer keyboard and monitor, and the second using the PPE board for input. Suggestions: outline and flow chart before coding, use subroutines, and make sure the subroutines function as program before converting them to subroutines.

Pre-Lab Work:
1. Create a flow chart for each program. Create .ASM files for each program (type the programs into a text editor program like “Notepad” or “Textpad” and save them with an “.ASM” extension).

Laboratory Work:
1. Demonstrate the programs, have your updated pre-lab Flow Charts, Listing files, demonstrate the programs.

Extra Credit:
1. Add floating point numbers to the program. Document with flow chart, commented code, and demonstrate to your lab instructor.
The Parallel Port Exerciser (PPE)

### PPE description

<table>
<thead>
<tr>
<th>Output Port @ 0378h</th>
<th>(D7 - D0)</th>
<th>Rows</th>
<th>Keypad</th>
</tr>
</thead>
<tbody>
<tr>
<td>(for D3 - D0)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>D0</td>
<td></td>
<td>1 2 3</td>
<td></td>
</tr>
<tr>
<td>D1</td>
<td></td>
<td>4 5 6</td>
<td></td>
</tr>
<tr>
<td>D2</td>
<td></td>
<td>7 8 9</td>
<td></td>
</tr>
<tr>
<td>D3</td>
<td></td>
<td>* 0 #</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Input Port @ 0379h</th>
<th>(S7 S6 S3 S4 S5)</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>^ ^ ^ ^ ^</td>
<td></td>
<td></td>
</tr>
<tr>
<td>white</td>
<td>red</td>
<td>1 2 3</td>
<td></td>
</tr>
<tr>
<td>button</td>
<td>button</td>
<td>4 5 6</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>7 8 9</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>* 0 #</td>
<td></td>
</tr>
</tbody>
</table>

Code to Poll the Key Pad for #1 or #2 or #3

```plaintext
;*
;* Test program for PPE board
;*

cseg segment 'code'
assume cs:cseg, ds:cseg, ss:cseg, es:cseg

ORG 0100h
START: MOV DX,0378h ;setup output port
       MOV AL,01h ;looking for 1,2,3
       OUT DX,AL ;enable the 1,2,3 row
       MOV DX,0379h ;setup input port
       IN  AL,DX ;get input data
       MOV AH,AL ;save the key data
       CMP AL,3Fh ;Nothing Pressed
       JZ  START
       AND AL,08h ;looking for #1 pressed in bit D3
       JZ  ONE
       MOV AL,AH ;get saved input key data
       AND AL,10h ;looking for #2 pressed in bit D3
       JZ  TWO
       MOV AL,AH ;get saved input key data
       AND AL,20h ;looking for #3 pressed in bit D3
       JZ  THREE
       JMP START

ONE:  MOV BL,01h
       JMP BYE

TWO:  MOV BL,02h
       JMP BYE

THREE: MOV BL,03h
       JMP BYE

BYE:   mov ah, 4ch
        INT 21

cseg ends
end start
```