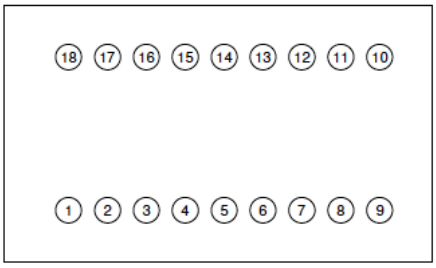
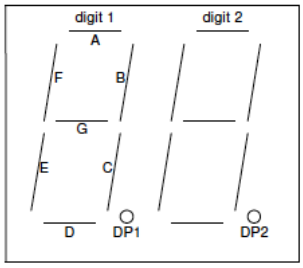
Lab #9 - pre lab

1) Assume pin 0 of an 8052 port is attached to LED segment A, pin1 is attached to segment B, and so on. Determine the byte that needs to be copied to the port to generate 0-9, A-F, and a blank display

0	
1	
2	
3	
4	
5	
6	
7	
8	
9	
A	
B	
C	
D	
E	
F	
blank display	





digit 1 segment E	1	18	digit 1 segment F
digit 1 segment D	2	17	digit 1 segment G
digit 1 segment C	3	16	digit 1 segment A
DP1	4	15	digit 1 segment B
digit 2 segment E	5	14	digit 1 common (+)
digit 2 segment D	6	13	digit 2 common (+)
digit 2 segment G	7	12	digit 2 segment F
digit 2 segment C	8	11	digit 2 segment A
DP2	9	10	digit 2 segment B

2) Given the byte values you found in problem 1, create pseudo-code for a C function that will display the passed in number on the display. Assume a 7-segment display is attached to a port defined by `#define SEG7_PORT`. If the number is above 15, it should blank the display. Hint: you can do this with 5 lines of actual code (or less?).

```
void displayNumber(unsigned char num)
```

3) Assume you have the following function `char getKeypad()` that returns a debounced keypress from the 4x4 keypad. The function returns decimal values as follows: 0 for a keypress of 0, 1 for a keypress of 1, ..., 10 for A, 11 for B, 12 for C, 13 for D, 14 for #, and 15 for *. The function returns -1 if no key was pressed.

Write pseudo-code for a main method that behaves as follows:

a) When a key is pressed, copies the return value of `getKeypad()` to the P0 LEDs.
b) If 0-9, A, B, C, D, or # is hit, adds the number to a running total
c) Displays the running total on the 7-segment display by calling `displayNumber()`
d) If the * key is hit, resets the total to 0.

4) Before writing the function `char getKeypad()`, you decide to write a simpler function `char getKeypadDown()`. This function does the row/column scanning of the keypad and returns a decimal value 0-15 depending on which key is currently pressed. If multiple keys are pressed, it returns the value for the first key it finds. If no key is pressed, it returns -1.

Note that this function does not need to wait for key release or debounce. Write pseudo-code for the `char getKeypadDown()` function.

5) You now are ready to write the function `char getKeypad()`. This function needs to determine what key (if any) has been pressed for a sufficient period of time to be a legitimate keypress (i.e. it must handle switch bounce). After a key has been down for long enough, it qualifies as a press but your function should still wait for the key to be released before returning.

Hint: you could use the power savings idle mode to perform the sleep. You can assume you have a `void sleep()` function that sleeps for 0.01s as I showed in lecture. Give the pseudo-code for the `char getKeypad()` function.