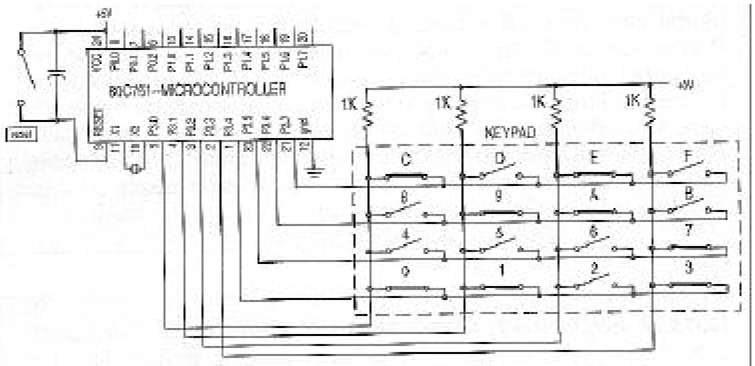
CpE213 HW7 - Due Nov. 11 at 11am.

Name____________________

1. (10 Points) The figure below shows a keypad connected to the 8051. We would like set a bit variable `pressed`=1 if any of the keys: D, 9, 5, or 1 are pressed and set `pressed`=0 otherwise. Write a segment of C-code to do this. Use as little code as possible (I was able to do it using 2 statements). If you can't be efficient, write something that works. If you think you need partial credit, explain your answer. Assume the keypad hasn't been scanned yet.



Keypad schematic

2. (15 Points) Write a complete C-program which does the same as the following assembly code:

```
BSEG AT 22H
        x: DS 2
mydata segment data
        y: DS 1 ; not used – declare anyway.
CSEG AT 0000H
        MOV x, #42H
        MOV A, #2AH
        CLR C
        RLC A
        MOV DPTR, #5280H
        MOVX @DPTR,A
        SETB 12H
```

3. (12 Points) For which of the following situations should you write your code in C or in ASM and for what situations does it not matter which language you use. (Mark the correct answer with an X).

	C	ASM	Doesn’t Matter
A. A hardware driver with very precise timing requirements - on the order of microseconds	______	______	______
B. A short task which is executed rarely	______	______	______
C. An application where speed or code size doesn’t matter	______	______	______
D. Your program is a few milliseconds too slow. One particular function is executed 90% of the time. That segment should be written in	______	______	______
E. Safety is critical, so the program should be easy to understand and debug.	______	______	______
F. Your code must be out the door yesterday. Your boss tells you to get it out fast at any cost. You write your code in	______	______	______

4. (24 Points) Say we have set aside space for an array, blah, containing 42 elements and we have created a 1-byte variable “i” in internal data space. Write a few instructions of ASM code (not the whole program) to perform the C-code operation **blah[i] = x** when:

a. (12 Points) blah is an array of char in internal data memory and x is a char in internal data memory

b. (12 Points) blah is an array of int in internal data memory and x is an int in external memory (If you could not get the code for part a, but understand how it might change for part b, explain how).

5. (24 Points) The following questions each ask you to predict a result after a segment of code has been executed. For each question, assume the contents of memory begin with the values below.

	Internal Direct Addr. mem.		Internal Indirect. Addr. mem.		External memory
0x81	23H	0x81	73H	0x5280	7AH
	:		:		:
0x42	27H	0x39	D0H	0x81	37H
	:		:		:
0x3A	22H			0x42	19H
	:				:
0x27	13H			0x3A	27H
	:				:
0x18	BAH				
	:				

a. (6 Points) Assume x is a 1 byte variable located at memory address 0x3A. After:

```
MOV R1, #42H
MOV x, @R1
```

x = ?

a) 42H   b) 27H   c) 22H   d) 7CH

e) 39H   f) 02H   g) 3AH   h) Unpredictable

b. (6 Points) Assume the declaration "sfr x = 0x81;" has been made at the top of the program. After:

```
x = (x&0xFE)|0x18;
```

x = ?

a) 98H   b) 1EH   c) 7AH   d) 3AH

e) invalid expn.   f) logical 1   g) 3E   h) Unpredictable

c. (6 Points) Assume the declaration "char idata x _at_ 0x81" has been made at the top of the program. After:

```
ACC =  x + XBYTE[0x42];
```

ACC = ?

a) 79H   b) 8CH   c) 5EH   d) 4AH

e) 3CH   f) 50H   g) 56H   h) Unpredictable

(Problem 5, continued)

d. (6 Points) After:

```
DSEG AT 27H
        x: DS 1
CSEG AT 00H
        MOV DPTR, #5280H
        MOVX A, @DPTR
        MOV X, A
        SETB 3AH
        CLRB 39H
```

x = ?

a) 7CH b) 7AH c) 15H d) 25H

e) 27H f) 13H g) 42H h) Unpredictable

6. (15 Points) Write an ASM program, complete with segment and variable declarations, to create a square wave on P1^5 with a 10% duty cycle (i.e. on 10% of the time and off 90% of the time) and a period of a “few” hundred machine cycles. You don’t have to be precise on the period, but you should be in the right ballpark. Define any constants using the EQU directive. Use relocatable segments.