CmpE213 – Test II

Name_____________________

Show all your work in the space provided. Answers with a simple “yes”, “no”, or a single number are typically incomplete and will not be given full credit. Answers in non-reduced form, like (a+sqrt(b))/c, are fine where appropriate. Good English on essay/short answer questions is required. ON MULTIPLE CHOICE QUESTIONS, IF YOU’RE NOT SURE DON’T GUESS – you will get points off for wrong answers.

1. (14 Points) Why is it less efficient in both speed and memory to use greater precision than is needed when writing your code. For example, would it take less time or code space to use a char compared to an int? Be explicit.

2. (14 Points) We would like to write a 1 out of bit 3 of port 2 without changing any other bits. Which of the following segments of code accomplish this task? Mark all that apply.

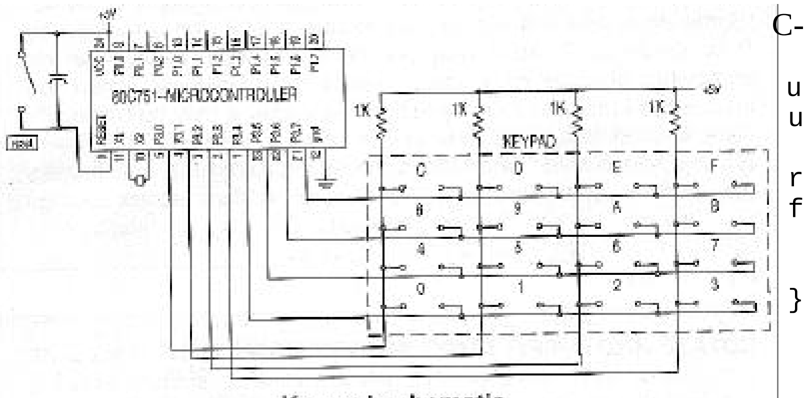
```
a) sbit mybit = 0xA3;   b) P2 = P2 & 0x08;    c) P2^2 = 1;
  mybit = 1;

d) P2 = 0x08;           e) P2 = P2|0x08;      f) P2 = P2 + 0x08;

g) sfr outport = 0xA0;  h) sbit abit = P2^3;
  outport^3 = 1;          abit = 1;
```

3. (14 Points) Why must we initialize the stack when writing an ASM program which uses functions? Why doesn’t the programmer have to initialize the stack when writing a C program with functions?

4. (15 Points) The following is a sketch of a keypad and a segment of C-code to read it (Like we did in class). Assuming the value 0x5280 ends up in variable `result`, show the state of the keys (i.e. open or closed) in the given schematic.



Keypad schematic

C-code:

```
unsigned char pat;
unsigned int result;

result = 0x0000;
for (pat=0x80;pat>0x08;pat>>1){
 P3 = ~pat;
 result =(result|(P3&0x0F))<<4);
}
```

5. (20 Points) Write a short program in C to send a square-wave out P3^2. Give the square wave a 25% duty cycle (i.e. on 25% of the time, off 75% of the time) and a period of a few hundred machine cycles (obviously, an approximation is fine here). Instead of writing directly to P3^2, assign an sbit variable to that location and use it in your code. Comment your code.

6. (23 Points) The following C code and ASM code are supposed to do the same thing, but the ASM code has (lots of) errors in it (The C code is correct). Identify the errors, explaining or correcting each error (just marking errors will not get you full credit). HINT: There is some important ASM code missing. (If you prefer, feel free to write your own ASM code instead of correcting mine).

```
/* C-code to do stuff */

int blah(char y);
char xdata myport _at_ 0x1234;
void main(void){
      int data x;

      while(1){
            if (P3^1){
                  x = blah(myport);
            }
      }
}
int blah(char y){
      if y==0
            return 0x5280;
      else
            return 0;
}
```

```
; ASM Code
mycode segment code
mydata segment idata
myxdata segment xdata

rseg myxdata
      myport: DS 1 at 1234H

rseg data
      x: DS 1
      y: DB 1
      stack: DS 20

CSEG mycode
      ; start main function
start:      JB B1H,start      ; jump if P3^1 set
            MOV R6,#myport    ; pass value in myport to blah()
            CALL blah         ; call function blah()
            MOV R0,x+1        ; return value to x
            MOV R1,x          ; return value to x
finish:     JB B1H,start      ; finish while loop

      ; start function blah
blah:       MOV y, R6         ; pass value into y
            MOV A,0H          ; clear A
            CJNE A,y,else     ; check if y==0
            MOV R0, 0         ;   return 0
            JMP stop
else:       MOV R0, #52       ; upper byte
            MOV R1, 80H       ; lower byte
stop:       JMP finish        ; return from function
```