EE 2310 Homework #7 – Complex SPIM Programs

Name: _____ CE ____ EE ____

For the problems given, develop MIPS programs that satisfy the specifications in the problem statement. Remember: On the homework due, email your NotePad files of the programs as attachments to the TA. Note: these are challenging programming problems, and deserve careful consideration and focus.

1. **Develop a program that arranges the series of decimal numbers shown in ascending order from smallest to largest. Use the space at the right for your program. Note the use of the ".word" directive to define multiple pieces of data with one data declaration.**

   When the numbers are ordered correctly, print them out as a single string from memory. Hint: You will need counters to determine when the compares are complete, and when you have printed out all the numbers. But note: The compare counter should only count eight (8), as you only do 8 compares. The print counter, however, should count nine (9), as you are printing 9 numbers!

```
#                    Order Program
#  Arranges numbers in ascending numerical order.

          .text
main:




          .data
nums:     .word 5,3,8,9,2,4,1,7,6
```

2. A factorial function can be defined as:

n! = n·(n-1)!, where 0!=1.

Then n! can be calculated as follows: A recursive procedure call (via jal) counts down from n to 1 to determine the number of loops necessary to calculate the n! result. Then, the jr ("procedure uncall") is used to produce a number of loops equal to the number of loops in the countdown to do the actual math. Using the definitions listed above for n! and 0!, construct a program that uses recursion to calculate n! in the manner suggested above.

The number will be input from the console. Note that the number must be 12 or less, since we are not doing floating-point calculations.

```
        .data
input:  .asciiz "Input integer (0-12): "
ans:    .asciiz " factorial is "
```

3. **In the space provided, write a program to count the number of consonants in the phrase "hello, world!\n" (as shown in the data statement).**

   Note that the "h" in the phrase is purposely left as a small letter in order to make your loop program a little easier to write (you don't have to worry about capital letters).

   When the program has calculated the number of consonants, use the answer header provided and then output the correct number. Hint: The total number of letters minus the vowels equals the number of consonants.

```
        .data
str:    .asciiz "hello, world!\n"
ans:    .asciiz "The number of consonants is "
```