For the problems given, develop MIPS programs that satisfy the specifications in the problem statement. Remember: To turn in this homework, email your NotePad files of the programs as attachments to the TA. Note: these are more advanced problems, and deserve careful consideration and focus.

1. (CLO 5—Assy Lang.) Develop a program that arranges the series of decimal numbers shown in ascending order from smallest to largest, then prints them out. If you wish, you can copy and paste the program in the space to your left so that you will have a printed record of it.

When the numbers are printed, put a carriage return/line feed between each, for neatness. Note 1: You will not need any counters, as the data provided has a final zero value, so that all you have to do is test for 0 and then quit. Note that the 0 is NOT to be printed out—it is not a value to be compared but merely an “edge” to tell you that the list is over.

Note 2: This must be a recursive program.

```assembly
# Problem 1: Arranging Numbers in Order
.text
main:

.data
num1: .word 53875
num2: .word 36429
num3: .word 88641
num4: .word 95437
num5: .word 25153
num6: .word 42398
num7: .word 0
```
2. (CLO 5—Assy Lang.) Compose a program to initialize an array with a series of numbers. This program will be very similar to one that we did in class, but with a twist or two. The array is to be 20 X 20 words, so you will need to use the “.space” command to initialize the array size. It should be 400 words (20 X 20), but remember, using the “.space” directive, you have to reserve bytes.

In this case, you are going to load each row of the array with slightly different numbers. You will be loading rows 0-19, and columns 0-19, or 20 X 20. Remember that the first row is 0! For even rows (including 0, that is, 0, 2, 4, 6, etc.), load the numbers 1-20 consecutively in the 20 columns. For odd rows (1, 3, 5, 7, etc.) load the numbers 20-1 consecutively, that is, in reverse order.

Thus, you will load even rows with the sequence 1-20, and odd rows with the sequence 20-1.

As in problem one, you can print out and paste a copy of the program in the space at the right for your records.

# Array Program -- Loads a 20 Word X 20 Word Array With Numbers
3. (CLO 5—Assy Lang.)
Compose a program to arrange the characters in “Hello, world!\n” in numerical order, smallest value to largest, and print out the result. That will of course mean that the string of characters looks more or less like gibberish, but the characters will be in proper numerical order, smallest to largest.

Note that this requires a recursive program to be done properly. As for the other two problems in this set, you can paste a copy of the program to the right for your records, if you wish.

```
.data
str: .asciiz "Hello, world!\n"
```