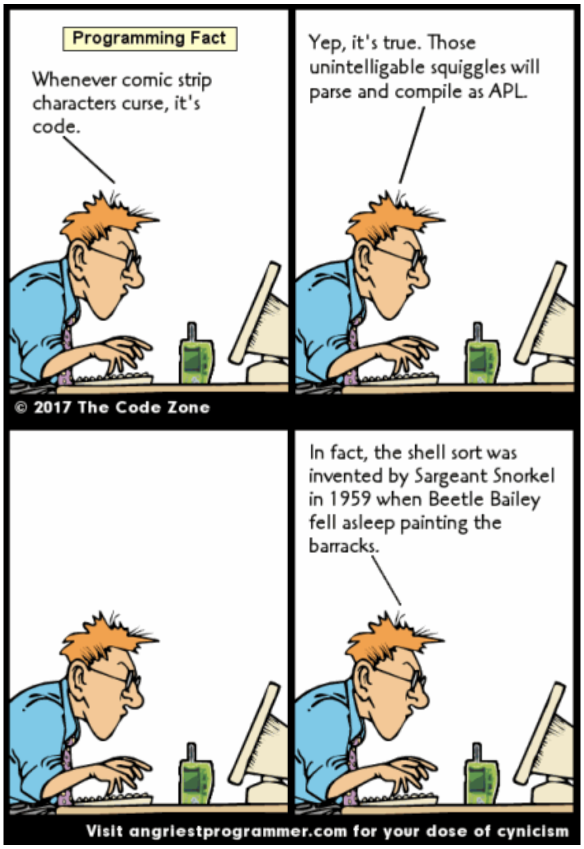
CS 218 – Homework, Asst. #7

Purpose: Write a simple assembly language program to sort a list of numbers. Learn to use addressing modes, arithmetic operations, and control instructions.
Due: Tuesday (2/13)
Points: 100

Assignment:

Write a simple assembly language program to sort a list of integer numbers into ascending (small to large) order. Additionally, find the minimum, median, maximum, sum, and average of the list. You should find the minimum and maximum after the list is sorted (i.e., min=list[0] and max=list[len-1]. *Note*, for an odd number of items, the median value is defined as the middle value. For an even number of values, it is the integer average of the two middle values. The median must be determined *after* the list is sorted.


Programming Fact
Whenever comic strip characters curse, it's code.
Yep, it's true. Those unintelligable squiggles will parse and compile as APL.
© 2017 The Code Zone
In fact, the shell sort was invented by Sargeant Snorkel in 1959 when Beetle Bailey fell asleep painting the barracks.
Visit angriestprogrammer.com for your dose of cynicism

To sort the numbers, use the following Shell Sort[1] algorithm. You must use the below Shell Sort algorithm (i.e., do not use a different sort). The algorithm assumes array index's start at 0. ***Submissions not based on this algorithm will not be scored***.

```
h = 1;
while ( (h*3+1) < length ) {
   h = 3 * h + 1;
}

while ( h>0 ) {
   for (i = h-1; i < length; i++) {
       tmp = lst[i];
       j = i;
       for ( j=i; (j >= h) && (lst[j-h] > tmp); j = j-h) {
          lst[j] = lst[j-h];
       }
       lst[j] = tmp;
   }
   h = h / 3;
}
```

All data must be treated as ***unsigned*** integers. As such, the MUL and DIV instructions and JA/JAE/JB/JBE comparisons should be used (not the IDIV and IMUL instructions or the JG/JGE/JL/JLE comparisons). Do not change the provided data types/sizes.

The program should display the minimum, maximum, median, sum, and average to the screen in base-36. Use the provided main, which includes the print routines. You will need to add the macro code from the previous assignment.

[1] For more information, refer to: http://en.wikipedia.org/wiki/Shell_sort

Submission:

When complete, submit:

- A copy of the ***source file*** via the class web page (assignment submission link) by class time (Section 1, 2:30 PM and Section 2, 5:30 PM). Assignments received after the due date/time will not be accepted.

Data Declarations:

Refer to the provide main for the provided data declarations.
As necessary, you can define additional variables.

Integer to Base-36 Macro:

This assignment uses the integer to base-36 conversion macro from assignment #6. The provided main includes a place to cut-and-paste the code from the assignment #6 macro into the assignment #7 template. The macro is used, along with some other provided macro's, to display output to the screen (as shown below).

Example Output:

The results, as displayed to the screen, would be as follows:

```
------------------------------------------------------
CS 218 - Assignment #7
Shell Sort

Minimum:             1
Maximum:           7PR
Median:             IE
Sum:              C3L1
Average:           1G9
```

Note, since this program displays output to the screen, it can be executed without the debugger.

Debugging Tips

- Use comments!!
- Follow the algorithm directly (do not attempt to optimize).
- Comment each part of the algorithm (so you can match the algorithm to the appropriate subset of code).
- Develop a debugger input file first (based on previous ones) carefully verifying the debugger commands based on the specific data types.
- You can temporarily change the array length to a smaller number (i.e., 5-10) for testing.