

CS 218 – Assignment #7

Purpose: Write a simple assembly language program to sort a list of numbers. Learn to use addressing modes, arithmetic operations, and control instructions.

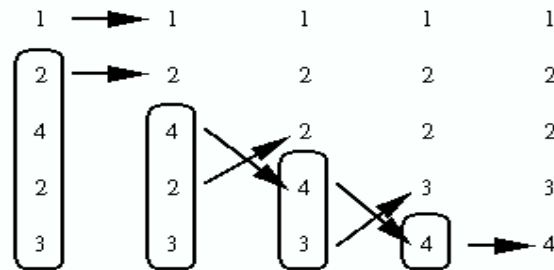
Due: Thursday (6/20)

Points: 100

Assignment:

Write a simple assembly language program to sort a list of signed integer numbers into ascending (small to large) order. Additionally, find the minimum, median, maximum, sum, and average of the list. You can find the minimum and maximum after the list is sorted (i.e., $\text{min}=\text{list}[0]$ and $\text{max}=\text{list}[\text{len}-1]$). 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. You should write the code for both even and odd length lists as it will be used in the next assignment.



To sort the numbers, use the following bubble sort¹ algorithm:

```
for ( i = (len-1) to 0 ) {  
    swapped = false  
    for ( j = 0 to i-1 )  
        if ( lst(j) > lst(j+1) ) {  
            tmp = lst(j)  
            lst(j) = lst(j+1)  
            lst(j+1) = tmp  
            swapped = true  
        }  
    if ( swapped = false ) exit  
}
```

You **must** use the above Bubble Sort algorithm (i.e., do **not** use a different sort). *Note*, the algorithm assumes array index's start at 0. As necessary, you can define additional variables.

Submissions not based on this algorithm will not be scored.

All data must be treated as **signed** integers (i.e., negative numbers). As such, the IMUL, IDIV, and CDQ instructions should be used (not the DIV and/or MUL). Do not change the provided data types/sizes.

Submission:

When complete, submit:

- A copy of the **source file** via the class web page.

Assignments received after 1:00 PM will not be accepted.

¹ For more information, refer to: http://en.wikipedia.org/wiki/Bubble_sort

Data Declarations:

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

Integer to Octal Macro:

This assignment uses the integer to octal 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 the provided print string macro, to display output to the screen (as shown below).

Example Output:

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

```
ed@ed-vm% ./ast7
CS 218 - Assignment #7

List Statistics:

List Minimum:      -21645
List Median:       +2144
List Maximum:      +20653
List Sum:           +1664046
List Average:      +3122
```

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.