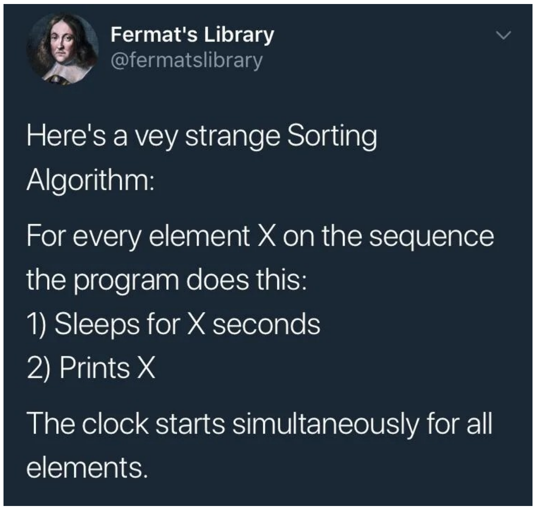
CS 218 – Assignment #7

Purpose: Write a simple assembly language program to sort a list of numbers. Learn to use addressing modes, arithmetic operations, and control instructions.
Due: Tuesday (2/18)
Points: 100

Assignment:

Write a simple assembly language program to sort an array of 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., min=array[0] and max=array[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 arrays as it will be used in the next assignment.



Fermat's Library
@fermatslibrary

Here's a vey strange Sorting Algorithm:

For every element X on the sequence the program does this:
1) Sleeps for X seconds
2) Prints X

The clock starts simultaneously for all elements.

Use the following selection[1] sort algorithm:

```
begin
    for  i = 0 to len-1 {
        small = arr[i]
        index = i
        for  j = i to len-1 {
            if (arr[j] < small) {
                small = arr[j]
                index = j
            }
        }
        arr[index] = arr[i]
        arr[i] = small
    }
end_begin
```

You ***must*** use the above selection sort algorithm above (i.e., do **not** use a different sort). *Note*, the algorithm assumes array indicies start at 0. As necessary, you can define additional variables. ***Submissions not based on this algorithm will not be scored***.

The program will display the minimum, maximum, median, sum, and average to the screen. Use the provided main, which includes the print routines. All data must be treated as ***unsigned*** integers. As such, the MUL and DIV instructions should be used (not the IDIV and/or IMUL) and the JA/JAE/JB/JBE comparisons should be used (not the JG/JGE/JB/JBE). Do not change the provided data types/sizes.

Submission:

When complete, submit:

- A copy of the ***source file*** via the class web page (assignment submission link) by 11:59 PM. Assignments received after the due date/time will not be accepted.

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

Data Declarations:

Refer to the class web page the provided template. As necessary, you can define additional variables.

Example Output:

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

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

Minimum:           9
Maximum:       11217
Median:         1572
Sum:          819002
Average:        2730

ed-vm%
```

Note, since this program displays output to the screen, it can be executed without the debugger (as shown above). You will still need to use the debugger script to show that the data is sorted.

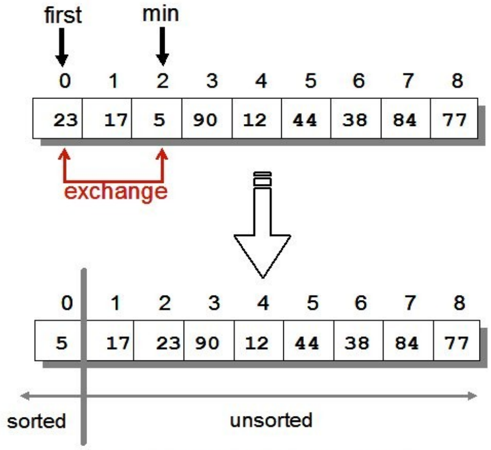
Debugging Tips

- Use comments!! Specifically, comment each part of the algorithm (so you can match the algorithm to the appropriate subset of code).
- Follow the algorithm directly (do not attempt to optimize).
- Develop a debugger input file first (based on previous ones) carefully verifying the appropriate 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.

Selection Sort Algorithm Overview:

Below is a summary of the selection sort algorithm (for one iteration of the outer loop):

1. Find the smallest element in the list.
2. Exchange the element in the first position and the smallest element. Now the smallest element is in the first position.
3. Repeat Step 1 and 2 with the list having one less element (i.e., the smallest element is discarded from further processing).



This is the result of one pass.