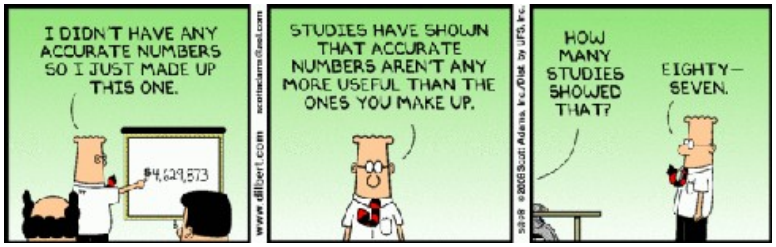
CS 218
Homework, Asst. #8

Purpose:	Learn assembly language procedures. Additionally, become more familiar with program control instructions, procedure handling, and stacks.
Due:	Monday (6/20)
Points:	125

Assignment:



Write four simple assembly language functions to provide some statistical information as described below. You will be provided a main that calls the following functions (for each set of data).

- Write a void function, **cocktailSort()**, to sort the passed array of numbers into descending order (large to small). You **must** use the cocktail sort algorithm from assignment 7, modified to sort in descending (large to small) order.

- Write a void function, **cubeAreas()**, to calculate the area of each cube in a series of cube sides.

- Write a void function, **cubeStats()**, that given an array of integer cube areas, finds the minimum, maximum, sum, integer average, sum of numbers evenly divisible by 4.

- Write an integer function, **iMedian()**, to compute and return the integer median for a list of numbers. *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. A 32-bit integer function returns the result in ***eax***.

- Write an integer function, **mStatistic()**, to compute the ***m***-statistic for a list of numbers. The formula for the ***m***-statistic is as follows:

$$mStat = \sum_{i=0}^{len-1} \left[\left(list[i] - median \right)^2 \right]$$

  The function must call the **iMedian()** function to find the integer median. A 64-bit integer function returns the result in ***rax***. *Note*, due to the data sizes, the summation must be performed as a quad-word.

All data should be treated as *signed* integers (IMUL, IDIV, and CDQ instructions, etc.). The functions must be in a separate assembly file. The files will be assembled individually and linked together.

Submission:
When complete, submit:

- *A copy of the **source file** for the functions (not the provided main) via the class web page start of class. Assignments received after the allotted time will not be accepted!*

Updated Linking Instructions

You should use the provided script file, **asm8**, to perform the assemble and link commands. For example, assuming the provided main is named **ast8main.asm** and the procedures file is named **ast8procs.asm** the following command;

```
ed-vm% ./asm8 ast8main ast8procs
```

will produce the executable file **ast8main** (which can be executed by typing **./ast8main**). You can change the file names as desired. *Note*, only the functions file will be submitted. The submitted functions file will be assembled (as noted above) with the provided main.

Refer to the text for more information regarding functions, controlling program execution, and finding logic errors.

Provided Data Sets:

Do not change the data types of the provided data. You may define additional variables as required. For example, given the below provided data set:

```
; -----
cSides1       dd      21,  27,  10,  22,  31
              dd      13,  12,  17,  19,  20
              dd      24,  11,  14,  30,  33
              dd      27,  34,  23,  37,  40
              dd      38,  18,  15,  25,  16
              dd      26,  39,  36,  13
len1          dd     29
min1          dd     0
med1          dd     0
max1          dd     0
sum1          dd     0
ave1          dd     0
fourSum1      dd     0
mStat1        dq     0
```

The results for data set #1 are shown for reference:

```
-------------------------------------
display data set #1

0x6015d0:  9600       9126       8664       8214
0x6015e0:  7776       6936       6534       5766
0x6015f0:  5400       4374       4374       4056
0x601600:  3750       3456       3174       2904
0x601610:  2646       2400       2166       1944
0x601620:  1734       1536       1350       1176
0x601630:  1014       1014       864        726
0x601640:  600

len:                  0x6010ac:  29
min:                  0x6010b0:  600
max:                  0x6010b8:  9600
med:                  0x6010b4:  3174
sum:                  0x6010bc:  113274
ave:                  0x6010c0:  3906
fourSum:              0x6010c4:  55962
mStat:                0x6010c8:  237465720
```