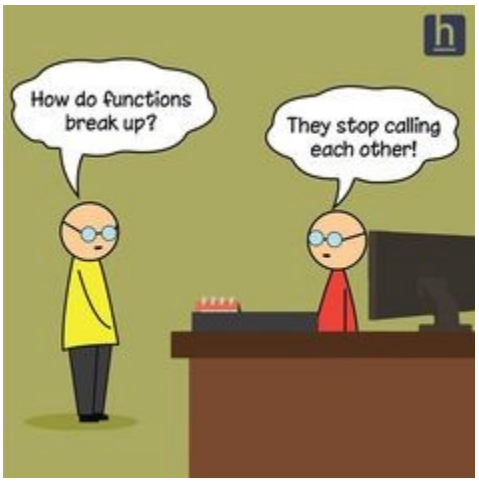
CS 218 – Assignment #8

Purpose: Learn assembly language functions. Additionally, become more familiar with program control instructions, function handling, and stack usage.
Points: 125

Assignment



Write a series of simple assembly language functions as described below. You will be provided a main function that calls the following functions (for each set of data).

- Write a value returning integer function, **lstEstMedian()**, to return the estimated median of a list of numbers (prior to sorting as done in assignments #4 and #5). The estimated the median is computed by summing the first, last, and two middle values and dividing by 4 for even length lists and 3 for odd length lists. The integer function returns the double-word value in ***eax***.
- Write a void function, **combSort()**, to sort the numbers into descending order (large to small). You **must** use the comb sort algorithm (from assignment 7). You will need to modify the algorithm to change the sort order (from ascending to descending).
- Write a value returning function, **lstSum()**, to find the sum for a list of numbers.
- Write a value returning function, **lstAverage()**, to find the average for a list of numbers. The function must call the **lstSum()** function to find the sum of the passed list.
- Write a value returning function, **lstMedian()**, to find the statistical median for a sorted list of numbers. The function should assume the list is sorted. For an odd number of items, the statistical median value is defined as the middle value. For an even number of values, it is the integer average of the two middle values.
- Write a void function, **lstStats()**, to find the sum, average, minimum, maximum, and median for a list of numbers. The function must call the **lstSum()**, **lstMedian()**, and the **lstAverage()** functions to find the median and average of the passed list.
- Write a value returning function, **lstKurtosis()**, to compute the kurtosis[1] statistic for the data set. The kurtosis formula is as follows:

$$kurtosis = \frac{\sum_{0}^{n-1} \left(x[i] - \bar{x}\right)^4}{\sum_{0}^{n-1} \left(x[i] - \bar{x}\right)^2}$$

Note, must perform the summation for the dividend (top) as a quad value. The $\bar{x}$ refers to the average. If the divisor (bottom) is 0, do not perform the divide and the final result should be set to 0.

1 For more information, refer to: https://en.wikipedia.org/wiki/Kurtosis

Assemble and Linking Instructions

You will be provided a main function (**`ast8main.asm`**) that calls the functions. Your functions should be in a separate file (**`ast8procs.asm`**). The files will be assembled individually and linked together.

When assembling, and linking the files for assignment #8, use the provided **makefile** to assemble, and link. *Note*, **only** the functions file, **`ast8procs.asm`**, will be submitted. The submitted functions file will be assembled and linked with the provided main. As such, do not alter the provided main.

Provided Data Sets

Refer to the provided main for the data sets. Do not change the data types of the provided data. You may define additional variables as required.

All data should be treated as ***signed*** integers (using the IMUL and IDIV instructions and JG/JGE/JL/JLE). The functions must be in a separate assembly file. The files will be assembled individually and linked together.

The results for data set #1 and #2 (partial) are shown for reference:

```
list1:
0x402000:  2968     2897      2785      2746
0x402010:  2660     2641      2562      2454
0x402020:  2411     2375      2259      2233
0x402030:  2123     1872      1871      1851
0x402040:  1730     1567      1542      1460
0x402050:  1351     1200      1195      1195
0x402060:  1187     1121      1120      1105
0x402070:  -1146    -1196     -1197     -1231
0x402080:  -2120    -2220     -2287     -2417
0x402090:  -2853

len1:      0x402094:  37

estMed1:   0x402098:  1387
min1:      0x4020a8:  -2853
med1:      0x40209c:  1542
max1:      0x4020ac:  2968
sum1:      0x4020a0:  37814
ave1:      0x4020a4:  1022

kStat1:    0x4020b0:  7763956



len2:      0x40226c:  110

estMed2:   0x402270:  1737
min2:      0x402280:  -2733
med2:      0x402274:  1261
max2:      0x402284:  2957
sum2:      0x402278:  122412
ave2:      0x40227c:  1112

kStat2:    0x402288:  8418135
```

Submission

- All source files must assemble and execute on Ubuntu with **`yasm`**.
- Submit source files
  - Submit a copy of the program source file via the on-line submission.
  - Only the functions file (**`ast8procs.asm`**) will be submitted.
- Once you submit, the system will score the project and provide feedback.
  - If you do not get full score, you can (and should) correct and resubmit.
  - You can re-submit an unlimited number of times before the due date/time (at a maximum rate of 5 submissions per hour).
- Late submissions will be accepted for a period of 24 hours after the due date/time for any given assignment. Late submissions will be subject to a ~2% reduction in points per an hour late. If you submit 1 minute - 1 hour late -2%, 1-2 hours late -4%, … , 23-24 hours late -50%. This means after 24 hours late submissions will receive an automatic 0.

Program Header Block

All source files must include your name, section number, assignment, NSHE number, and program description. The required format is as follows:

```
;  Name: <your name>
;  NSHE_ID: <your id>
;  Section: <4-digit-section>
;  Assignment: <assignment number>
;  Description: <short description of program goes here>
```

Failure to include your name in this format will result in a loss of up to 3%.

Scoring Rubric

Scoring will include functionality, code quality, and documentation. Below is a summary of the scoring rubric for this assignment.

Criteria	Weight	Summary
Assemble	-	Failure to assemble will result in a score of 0.
Program Header	3%	Must include header block in the required format (see above).
General Comments	7%	Must include an appropriate level of program documentation.
Program Functionality (and on-time)	90%	Program must meet the functional requirements as outlined in the assignment. Must be submitted on time for full score.