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).

HOW IS YOUR RESEARCH GOING?
UH, SLOWLY. I'VE GOTTEN THREE DATA POINTS IN THE LAST 60 DAYS.
LET ME SEE.
HMM, I THINK THERE'S A CLEAR TREND HAPPENING HERE.
DATA
JORGE CHAM ©2013
THERE IS??
YOUR CHANCES OF GRADUATING ARE ASYMPTOTICALLY APPROACHING ZERO.
DATA
WWW.PHDCOMICS.COM

- Write a value returning integer function, **estimatedMedia()**, to return the estimated median of a list of numbers (prior to sorting as done in assignments #4 and #5). For an even number of values, the estimated median will be computed by summing the two middle values and then dividing by 2. For an odd number of values, the estimated median will be the middle value. The integer function returns the value in ***rax***.

- Write a void function, **oddEvenSort()**, to sort the numbers into descending order (large to small). You **must** use the sort algorithm (from assignment 7). You will need to modify the algorithm to change the sort order (from ascending to descending).

- Write a void function, **listStats()**, to find the minimum, maximum, and median for an array of integer 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.

- Write a value returning integer function **estimatedSkew()**, to calculate an estimated skew value for a list of integer numbers. In probability theory and statistics, skewness is a measure of the extent to which a probability distribution "leans" to one side of the mean. To calculate the estimated skew[1], use the following formula:

$$skew = \frac{\sum_{i=0}^{len-1} \left((lst[i] - average)^2 \right)}{len * 3}$$

  Note, perform the summation and division using integer values. The function returns the result in ***eax***. *Note*, due to the data sizes, the summation ***must*** be performed as a quad-word.

[1] *Note*, the estimated skew formula has been simplified and thus will not provide a accurate measure for all data sets.

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 ***unsigned*** integers (using the MUL and DIV instructions and JA/JAE/JB/JBE). 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 #3 (partial) are shown for reference:

```
array1:
0x402000:          9999     3321     2346     2223
0x402010:          1427     1212     1160     1141
0x402020:          1137     1111     1110     475
0x402030:          216      150      141      120
0x402040:          90       61       33       33
0x402050:          30       30       27       23
0x402060:          22       22       20       20
0x402070:          19       18       13       12
0x402080:          12       10       3

len1:              0x40208c:         35
min1:              0x402090:         3
max1:              0x402094:         9999
med1:              0x402098:         61
sum1:              0x40209c:         27787
ave1:              0x4020a0:         793
estMed1:           0x4020a4:         12
skew1:             0x4020a8:         1044334


len3:              0x4023b0:         130
min3:              0x4023b4:         11124
max3:              0x4023b8:         99153
med3:              0x4023bc:         42786
ave3:              0x4023c4:         47166
sum3:              0x4023c0:         6131664
estMed3:           0x4023c8:         36169
skew3:             0x4023cc:         218880424
```

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.