**CS 218 – Assignment #8**

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

**Assignment:**
Write the assembly language procedures described below. You will be provided a main procedure that calls the following procedures/functions (for each set of data).

"Data don't make any sense,
we will have to resort to statistics."

- Write a void function, **shellSort()**, to sort the numbers into descending order (large to small). You **must** use the shell sort algorithm (from assignment 7) and modify the sort order.

- 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. *Note*, this function must call the **lstSum()** function.

- Write a void function, **basicStats()**, to find the minimum, median, maximum, sum, and average 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. This function must call the **lstSum()** and **lstAverage()** functions.

- Write a void function, **linearRegression()**, to compute the linear regression values ($b_0$ and $b_1$) for the two data sets. The linear regression formulas are as follows:

$$b_1 = \frac{\sum_{i=0}^{len-1} \left[ (x_i - \bar{x})(y_i - \bar{y}) \right]}{\sum_{i=0}^{len-1} (x_i - \bar{x})^2}$$

$$b_0 = \bar{y} - b_1 \bar{x}$$

*Note*, perform the summation and division using integer values. Due to the data sizes, the summation for the $b_1$ dividend (top) ***must*** be performed as a *quad-word*.

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

**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.

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

```
xList1:
0x402000:       5370    4780    4660    3510
0x402010:       3440    2890    2220    2110
0x402020:       1630    1250    -1310   -3120
0x402030:       -7410

yList1:
0x402034:       951120  921200  831100  741160
0x402044:       731150  631170  613213  542118
0x402054:       531110  431190  412130  -12110
0x402064:       -92140

len_1:          0x402068:       13

xMin_1:         0x40206c:       -7410
xMed_1:         0x402070:       2220
xMax_1:         0x402074:       5370
xSum_1:         0x402078:       20020
xAve_1:         0x40207c:       1540

yMin_1:         0x402080:       -92140
yMed_1:         0x402084:       613213
yMax_1:         0x402088:       951120
ySum_1:         0x40208c:       7232411
yAve_1:         0x402090:       556339

b0_1:           0x402094:       425439
b1_1:           0x402098:       85
```

**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: <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. |