

## CS 218 – Assignment #9

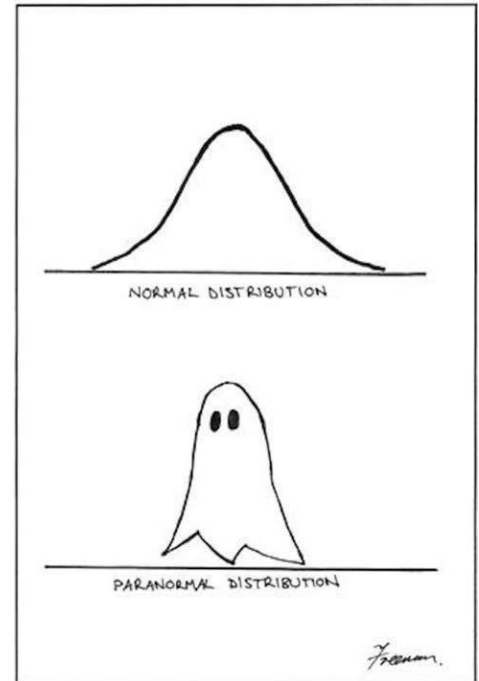
Purpose: Learn assembly language functions and standard calling convention. Additionally, become more familiar with program control instructions, high-level language function interfacing.

Points: 170

### Assignment:

Write the assembly language functions described below. You will be provided a C++ main program that calls the functions from assignment #8.

- Void function, **gnomeSort()**, to sort the numbers into ascending order (small to large).
- Void function, **basicStats()**, to find the minimum, median, maximum, sum, and average for a list of numbers. This function must call the **listMedian()** function.
- Value returning integer function, **listMedian()**, to find and return an integer median of a sorted list of numbers. 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. An integer function returns the value in **rax**.
- Value returning, double function, **corrCoefficient()**, to compute the correlation coefficient<sup>1</sup> for the two data sets. *Note*, a double function returns the value in **xmm0**.



In addition, write a function **readB13Num()** that will read an ASCII base-13 number from the user. The routine should use the system service for reading data from the keyboard (into a buffer), convert the ASCII base-13 input (from the buffer) into an integer, return the integer, and a status code. The number must be between MIN and MAX (inclusive). If the value is correct and within range, the function should return a status code (in **eax**) and, if successful, the numeric value (via pass-by-reference). The function must return one of the following status codes:

- SUCCESS (0) → Successful conversion and number within required range.
- BADNUMBER (1) → Invalid input entered (i.e., not a valid base-13 number).
- INPUTOVERFLOW (2) → User entry character count exceeds maximum length.
- OUTOFRANGE (3) → Valid base-13 number entered, but out of required range.
- ENDOFINPUT (4) → Return entered, no characters (for end of the input).

**All functions must use the stack for the storage of local variables.** No static variables should be declared. All data items are *unsigned* integers (MUL and DIV instructions). The functions must be in a separate assembly file. The files will be compiled/assembled individually and linked together.

<sup>1</sup> For more information, refer to: [http://en.wikipedia.org/wiki/Pearson\\_product-moment\\_correlation\\_coefficient](http://en.wikipedia.org/wiki/Pearson_product-moment_correlation_coefficient)

### **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.
  - Note, only the functions file (**ast9procs.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.
- Late submissions will be accepted for a period of 24 hours after the due date/time for any given lab. 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 10%.

### **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                      | 5%     | Must include header block in the required format (see above).  |
| General Comments                    | 10%    | Must include an appropriate level of program documentation.  |
| Program Functionality (and on-time) | 85%    | Program must meet the functional requirements as outlined in the assignment. Must be submitted on time for full score. |

### Example Execution:

The following is an example execution demonstrating various error handling:

```
ed@ubuntu~/cs218/ast9$
ed@ubuntu~/cs218/ast9$ ./main
-----
CS 218 - Assignment #9

Enter X Value (base-13): 101
Enter Y Value (base-13):      1ca
Enter X Value (base-13): 1cccccccccccc
Error, number of out of range. Please re-enter.
Enter X Value (base-13): 1ccc
Enter Y Value (base-13): 5
Enter X Value (base-13): d
Error, invalid number. Please re-enter.
Enter X Value (base-13): 5B
Enter Y Value (base-13):
bbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbbb
bbbbbbb
Error, user input exceeded length. Please re-enter.
Enter Y Value (base-13):      b
Enter X Value (base-13):  +4
Error, invalid number. Please re-enter.
Enter X Value (base-13):  4
Enter Y Value (base-13):  C51
Enter X Value (base-13):  1@3
Error, invalid number. Please re-enter.
Enter X Value (base-13):  1A3
Enter Y Value (base-13):  1aaaaaaaaaaaaa
Error, number of out of range. Please re-enter.
Enter Y Value (base-13):  1aa
Enter X Value (base-13):
-----
```

#### Program Results

##### Sorted X List:

4 76 170 302 4393

|             |      |
|-------------|------|
| X Minimum = | 4    |
| X Median =  | 170  |
| X Maximum = | 4393 |
| X Sum =     | 4945 |
| X Average = | 989  |

##### Sorted Y List:

5 11 309 335 2094

|             |      |
|-------------|------|
| Y Minimum = | 5    |
| Y Median =  | 309  |
| Y Maximum = | 2094 |
| Y Sum =     | 2754 |
| Y Average = | 550  |

Correlation Coefficient = 0.990301403774022

ed@ubuntu~/cs218/ast9\$

ed@ubuntu~/cs218/ast9\$