CS 218 – Assignment #9

Purpose: Learn assembly language functions and standard calling convention. Additionally, become more familiar with program control instructions, function handling, and stacks.
Due: Thursday (3/07)
Points: 150

YES, I GOT IT TO WORK!
I HAVE A POSITIVE RESULT!
SAMPLE SIZE: N = 1!
GRAD SCHOOL: GETTING EXCITED ABOUT INSIGNIFICANT SIGNIFICANCE.
a few hundred more to go!
JORGE CHAM © 2012
WWW.PHDCOMICS.COM

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

- Void function, **combSort()**, to sort the numbers into ascending order (small to large).
- Void function, **basicStats()**, to find the minimum, maximum, median, sum, and average.
- Value returning function, **intStdDev()**, to compute the integer standard deviation for a list of integers.
- Value returning function, **iSqrt()**, to return the integer square root of a passed value. *Note*, if the passed value is 0, the function should return 0.

In addition, write a function **readDozenalNumber()** that will read a signed ASCII duodecimal number from the user. The routine should use the system service for reading data from the keyboard (into a buffer), convert the Duodecimal/ASCII input (from the buffer) into an integer and return the integer. The number must be between the defined constants MIN_NUM and MAX_MAX (inclusive). If the entered input is valid and within range, the function should return the status of SUCCESS and the value by reference. When the end of input is received (a return with no characters on the line), the function should return a status of ENDOFINPUT (indicating no more input). If the entered input is invalid, the function should return NOSUCCESS. If the entered value is valid, but out of range, the function should return a status of OUTOFRANGE. If too many characters are entered, the function should return INPUTOVERFLOW.

All functions should use the stack for the storage of local variables. No static variables allowed!

All data items are *signed* integers (i.e., use IMUL and IDIV instructions as appropriate). The functions must be in a separate assembly file. The files will be assembled/compiled individually and linked together. Refer to the text, Chapter 12, for more information regarding functions. Refer to the text, Chapter 6, for more information regarding controlling program execution to find logic errors.

Submission:
When complete, submit:

- A copy of the ***source file*** via the class web page (assignment submission link) before 11:55 PM. ***Assignments received after the allotted time will not be accepted!***

Updated Compile, Assemble, and Linking Instructions
When compiling, assembling, and linking the files for assignment #9, use the provided compile, assemble, and link script file. *Note*, **only** the functions file will be submitted. The submitted functions file will be assembled (as noted above) with the provided main.

Testing

A script file to execute the program on a series of predefined inputs will be provided. *Note*, please follow the I/O examples. The test utility should be downloaded into an empty directory and the program executable placed in that directory. The test script, named **`a9tst`**, can be executed as follows:

```
ed-vm% ./a9tst main
```

The test script compares the program output to predefined expected output (based on the example I/O).

Example Execution:

The following is an example execution demonstrating various error handling:

```
ed-vm% ./main
--------------------------------------------------
CS 218 - Assignment #9

Enter Dozenal Value: +10
Enter Dozenal Value: +12
Enter Dozenal Value:     +13
Enter Dozenal Value:   +00000014
Enter Dozenal Value: +0000 15
Error, invalid number.  Please re-enter.
Enter Dozenal Value: +0000000000015
Enter Dozenal Value:           -0000016
Enter Dozenal Value:  +018
Enter Dozenal Value: +19
Enter Dozenal Value:   +00001x
Enter Dozenal Value: 1x
Error, invalid number.  Please re-enter.
Enter Dozenal Value: +999999999999
Error, number of out of range.  Please re-enter.
Enter Dozenal Value:  - 99
Error, invalid number.  Please re-enter.
Enter Dozenal Value: -9999999
Error, number of out of range.  Please re-enter.
Enter Dozenal Value:   +000001e
Enter Dozenal Value:    +0000000000000000000000000000000000000000000001
Error, input too long.  Please re-enter.
Enter Dozenal Value: +23
Enter Dozenal Value:

---------------------------------------------------------------
Program Results

Sorted List:
-18  12  14  15  16  17  20  21  22  23
27

Statistical Results:

              Length =            11
             Minimum =           -18
             Maximum =            27
              Median =            17
                 Sum =           169
             Average =            15
  Standard Deviation =            11

 ed-vm%
```