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.
Due:	Tuesday (2/28)
Points:	150

Assignment:

Write the assembly language functions described below. You will be provided a C++ main program that uses the functions.

"I can prove it or disprove it! What do you want me to do?"

- Write a value returning function, **listEstMedian()**, to find the estimated median for an unsorted list of numbers.
- Write a void function, **selectionSort()**, to sort the numbers into ascending order (small to large). You **must** use the selection sort algorithm and modify the sort order.
- Write a value returning function, **lstAverage()**, to find the average for a list of numbers.
- Write a void function, **listStats()**, to find the minimum, maximum, average, and median for a list of numbers. The function must call the **lstAverage()** functions to find the median and average of the passed list. The function should also calculate the integer percent error between the estimated median and the actual median.
- Write a value returning function, **betaValue()**, to compute the alpha-beta statistic for the data set.
- Value returning function, **rdTriNum()**, that will read an unsigned ASCII tridecimal number from the user. The routine must use the system service for reading data from STDIN (into a buffer) and perform the conversion (with the input from the buffer). Leading spaces should be ignored. If the input line is too long (> 50 characters), the input should be ignored and the function should return the appropriate status code. If no input (i.e., user typed enter and nothing else), the function should return false. The function must return one of the following status codes:
  - SUCCESS (0) → Successful conversion and number within required range.
  - NOSUCCESS (1) → Invalid input entered (i.e., not a valid tridecimal number).
  - OVERMAX (2) → The valid tridecimal number entered is > MAXNUM value (a provided constant).
  - INPUTOVERFLOW (3) → User entry character count exceeds maximum length.
  - 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 assembled individually and linked together.

Submission:

When complete, submit:

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

Testing

A script file to execute the program on a series of predefined inputs will be provided. 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 script file will require execute privilege (i.e., **`chmod +x a9tst`**). 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). *Note*, the **`ed@vm%`** is the prompt on my machine.

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 (**`asm9`**). *Note*, **only** the functions file will be submitted. The script file will require execute privilege (i.e., **`chmod +x asm9`**). The submitted functions file will be assembled and linked with the provided main. As such, do not alter the provided main.

Example Execution:

The following is an example execution demonstrating the error handling:

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

Enter values (return to end input).
Enter Value (tridecimal): 10
Enter Value (tridecimal): 1a
Enter Value (tridecimal): 1B
Enter Value (tridecimal): one
Error, invalid number. Please re-enter.
Enter Value (tridecimal): -10
Error, invalid number. Please re-enter.
Enter Value (tridecimal): aBc
Enter Value (tridecimal): five
Error, invalid number. Please re-enter.
Enter Value (tridecimal): ABCABCABCABCABCABC
Error, number above maximum value. Please re-enter.
Enter Value (tridecimal): 9999999999999999999999
Error, number above maximum value. Please re-enter.
Enter Value (tridecimal):
0000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
00000000004
Error, user input exceeded length, input ignored. Please re-enter.
Enter Value (tridecimal): 1Bc
Enter Value (tridecimal): +9
Error, invalid number. Please re-enter.
Enter Value (tridecimal):
--------------------------------------------------

Program Results

Sorted List:
    13      23      24     324    1845

      Length =               5
  Est Median =             120
     Minimum =              13
      Median =              24
     Maximum =            1845
     Average =             445
       Beta  =       902066648
 Err Percent =             400
ed-vm%
ed-vm%
```