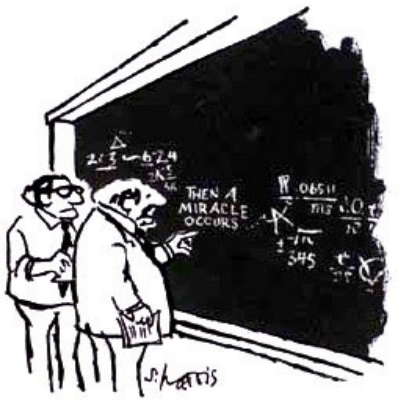
CS 218 – Assignment #9

Purpose: Learn assembly language functions and standard calling convention. Additionally, become more familiar with program control instructions, high-level language function interfacing.
Due: Wednesday (6/26)
Points: 150

Assignment:
You will be provided a C++ main program that calls the functions from assignment #8.



"I THINK YOU SHOULD BE MORE EXPLICIT HERE IN STEP TWO."

- Void function, **bubbleSort()**, to sort the numbers into ascending order (small to large) modified from the previous assignment.
- Void function, **cubeAreas()**, to calculate the area of each cube given a series of cube sides.
- Void function, **cubeStats()**, to find the minimum, maximum, sum, integer average, sum of numbers evenly divisible by 3 for a array of integer cube areas.
- Integer function, **iMedian()**, to compute and return the integer median 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.
- Integer long long function, **eStatistic()**, to compute and the ***e-***statistic for a list of numbers. The function must call the **iMedian()** function to find the integer median. *Note*, due to the data sizes, the summation must be performed as a quad-word.

In addition, write a function **readOctalNum()** that will read an octal/ASCII number from the user. The routine should use the system service for reading data from the keyboard (into a buffer), convert the ASCII binary input (from the buffer) into an integer, return the integer (via reference), and return a status code. The number must be ≥ MINNUMBER and ≤ MAXNUMBER (defined constants). The function should return the status code (in **rax**) and, if successful, the numeric value (via reference). The function must return one of the following status codes:

- EXIT_SUCCESS → Successful conversion and number within required range.
- EXIT_NOSUCCESS → Invalid input entered (i.e., not a valid binary number).
- OUTOFRANGEMIN → Valid binary number entered, but below minimum value.
- OUTOFRANGEMAX → Valid binary number entered, but above maximum value.
- INPUTOVERFLOW → User entry character count exceeds maximum length.
- ENDOFINPUT → 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, etc.). 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 functions ***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. *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. *Note*, the script file will require execute privilege (i.e., **`chmod +x asm9`**). The test script, named **`a9tst`**, can be executed as follows:

```
./a9tst ast9
```

The test script compares the program output to predefined expected output (based on the example I/O). As such, you should not change the provided error/status message strings.

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 and linked with the provided main. ***As such, you must not change the provided main!***

Example Execution:

The following is an example execution demonstrating various error handling:

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

Enter Value (octal): 1
Enter Value (octal): 2
Enter Value (octal): 3
Enter Value (octal): 4
Enter Value (octal): 5
Enter Value (octal): 6
Enter Value (octal): 7
Enter Value (octal): 8
Error, invalid number. Please re-enter.
Enter Value (octal): 9
Error, invalid number. Please re-enter.
Enter Value (octal): 10
Enter Value (octal): 1000000
Error, number above maximum value. Please re-enter.
Enter Value (octal): 11
Enter Value (octal): 1a
Error, invalid number. Please re-enter.
Enter Value (octal): 12
Enter Value (octal):
-------------------------------------------------

Program Results

Sorted Cube Areas List:
     6      24      54      96     150
   216     294     384     486     600

  Count       =            10
  Minimum     =             6
  Maximum     =           600
  Median      =           183
  Sum         =          2310
  Average     =           231
  Three Sum   =          2310
  e-statistic =        401418
```