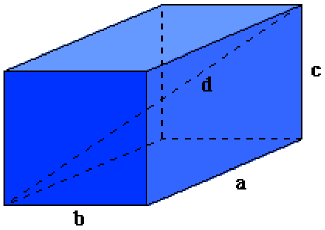
CS 218 – Assignment #2

Purpose: Become familiar with the MIPS stack and standard calling conventions.
Points: 75

Assignment:

Use the provided MIPS assembly language main program and write the following functions:

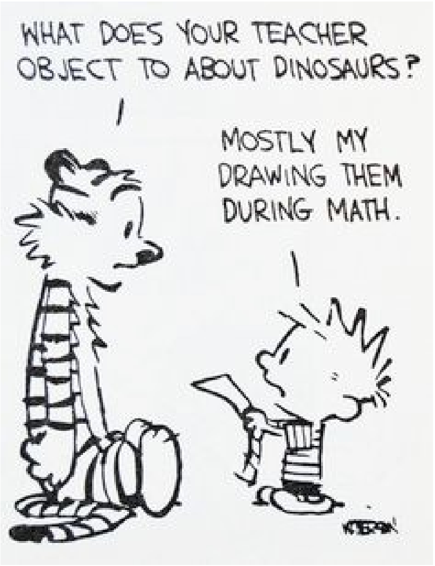
- Write a MIPS assembly language function, ***surfaceAreas()***, to calculate the surface areas for each of the rectangular parallelepipeds in a set of rectangular parallelepipeds. The formula for the surface of a rectangular parallelepiped is as follows:



$$\boldsymbol{surfaceAreas[n] \;=\; 2\left(asides[n]*bsides[n]+asides[n]*csides[n]+bsides[n]*csides[n]\right)}$$

- Write a MIPS assembly language function, ***bubbleSort()***, to sort a list of surface areas into ascending order (small to large). To sort the numbers, use the following bubble sort[1] algorithm:

```
for ( i = (len-1) to 0 ) {
     swapped = false
     for ( j = 0 to i-1 )
         if ( lst(j) > lst(j+1) ) {
             tmp = lst(j)
             lst(j) = lst(j+1)
             lst(j+1) = tmp
             swapped = true
         }
     if ( swapped = false ) exit
}
```


WHAT DOES YOUR TEACHER OBJECT TO ABOUT DINOSAURS?
MOSTLY MY DRAWING THEM DURING MATH.

You ***must*** use the above Bubble Sort algorithm (i.e., do **not** use a different sort). *Note*, the algorithm assumes array index's start at 0. As necessary, you can define additional variables. ***Submissions not based on this algorithm will not be scored***.

- Write a MIPS assembly language function, ***printAreas()***, to display the array of surface areas. The numbers should be printed five (5) per line, left justified (see example output).

- Write a MIPS assembly language function, ***surfaceAreasStats()***, that will find the minimum, median, maximum, sum, and floating point average. You should find the minimum, median, and maximum after the list is sorted. The average should be calculated as a floating point value.

- Write a MIPS assembly language function, ***printStats()***, to print the surface areas statistical information (minimum, maximum, median, sum, average) in the format shown in the example. The volumes should be displayed five (6) per line (they do not need to be justified). Display a tab (predefined in the template) before each volumes number.

[1] For more information, refer to: http://en.wikipedia.org/wiki/Bubble_sort

Submission:

- All source files must assemble and execute with QtSpim/SPIM MIPS simulator.
- Submit source file
  - Submit a copy of the program source file via the on-line submission
- 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 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 reduction of points.

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.

Example Output:

The program must display the results to the console window. The output should look something like the following (with all of the correct answers displayed for all data sets):

```
Assignment #2
Surface Areas Program

---------------------------
Data Set #1
Length: 20

 Unsorted Surface Areas:

          6698      6184      5622      6190      5608
          6766      6184      7038      6904      6758
          5008      174       18480     5240      6976
          7864      4760      6118      88        183800

 Sorted Surface Areas:

          88        174       4760      5008      5240
          5608      5622      6118      6184      6184
          6190      6698      6758      6766      6904
          6976      7038      7864      18480     183800

 Surface Areas Min = 88
 Surface Areas Med = 6187
 Surface Areas Max = 183800
 Surface Areas Sum = 302460
 Surface Areas Ave = 15123.00000000

---------------------------
Data Set #2
Length: 77

 Unsorted Surface Areas:

          304188    345742    380984    344290    336520
          339660    318010    343150    322120    311830

     [...truncated for space...]
```

Note, the example output above may appear slightly different than the console output on the screen.