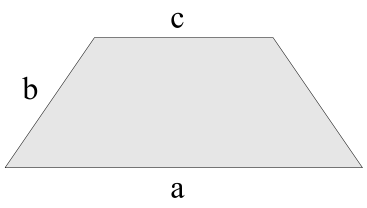
CS 218 – MIPS Assignment #3

Purpose: Become familiar with the MIPS stack and function calling convention.
Points: 80

Assignment

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

- Write a MIPS void function, ***trapAreas()***, to calculate the area of each trapezoid in a series of trapezoids. The formula for the trapezoid areas is as follows:



$$areas[i] = \left(heights[i] \; \frac{aSides[i] + cSides[i]}{2} \right)$$

- Write a MIPS void function, ***oddEvenSort()***. An Odd/EvenSort is a relatively simple sorting algorithm, developed originally for use on parallel processors. To sort the numbers, use the following Odd/Even Sort algorithm:

```
function oddEvenSort(list) {
     bool sorted = false;
     while (!sorted) {
          sorted = true;
          for (var i=1; i < len-1; i+=2) {
                    if (list[i] > list[i+1]) {
                              swap(list, i, i+1);
                              sorted = false;
                    }
          }
          for (var i=0; i < len-1; i+=2) {
                    if (list[i] > list[i+1]) {
                              swap(list, i, i+1);
                              sorted = false;
                    }
          }
     }
}
```

You ***must*** use the above sort algorithm above (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***.

How do you know if a trapezoid is friendly?
It has acute angles.

- Write a MIPS value returning function, ***estMedian()***, to compute the estimated median of the unsorted trapezoid areas array. If the list length is even, the estimated median is computed by summing the two middle values and then dividing by 2. If the list length is odd, the estimated median is the middle value.

- Write a MIPS void function, ***trapStats()***, that will find the minimum, median, maximum, sum, and average of the trapezoid areas. The function is called after the list is sorted. The sum and average should be calculated as a floating point value.

- Write a MIPS assembly language function, ***showTrapStats()***, to print the list and the statistical information (minimum, maximum, median, estimated median, sum, average) in the format shown in the example. In addition, the function should compute the difference between actual median and the estimated median. The formula for percentage change is as follows:

$$pctDiff = \frac{median_{est} + median_{real}}{median_{real}}$$

  The numbers should be printed 7 per line (see example).

Submission

When complete, submit:

- A copy of the **source file** via the class web page before class time.

Example Output

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

```
MIPS Assignment #3
Trapezoid Areas Program


*******************************************************************
Data Set #1
Length: 15

Trapezoid Areas:
     3600     3640     3723     3744     3848     3848     3876
     3888     4028     4070     4256     4332     4350     4425
     4524

 sum = 60152.00000000
 ave = 4010.13330078
 min = 3600
 med = 3888
 max = 4524
 est med = 4332
 pct diff = 2.11419749


*******************************************************************
Data Set #2
Length: 75
```

Submission

- All source files must assemble and execute with QtSpim/SPIM MIPS simulator.
- Submit source files
  - 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.
- Late submissions will be accepted for a period of 24 hours after the due date/time for any given assignment. 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 3%.

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.
Correct Sort Algorithm	-	Failure to use the provided sort algorithm 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.