

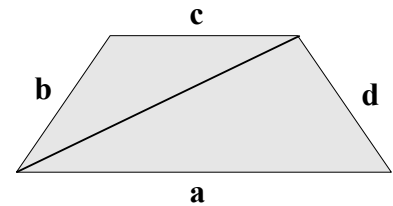
CS 218 – Assignment #3

Purpose: Become familiar with the MIPS stack, function calling, and the MIPS standard calling convention.

Points: 80

Assignment:

Write a MIPS assembly language program to calculate the diagonal of each trapezoid¹ in a series of trapezoids. Use the provided MIPS main program and develop the following functions:



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

$$diags[i] = \sqrt{\frac{a[i] \times b[i]^2 - a[i]^2 \times b[i] - a[i] \times c[i]^2 + b[i] \times d[i]^2}{b[i] - a[i]}}$$

Note, the routine is must call the *gnomeSort()* function for the diagonals array before returning.

- Write a MIPS assembly language function, *gnomeSort()*, to sort the volumes into ascending order (small to large). To sort the numbers, use the following Gnome sort² algorithm:

```
gnomeSort(a[0..size-1]) {  
    i := 1  
    j := 2  
    while (i < size) {  
        if (a[i-1] >= a[i]) {  
            i := j  
            j := j + 1  
        } else {  
            swap a[i-1] and a[i]  
            i := i - 1  
            if (i = 0) i := 1  
        }  
    }  
}
```

You **must** use the above gnome 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.

- Write a void MIPS function, *diagonalsStats()*, that will find the minimum, maximum, median, and float average of the diagonals array.
- Write a value returning MIPS function, *findSum()*, to find the sum of an array.

¹ For more information, refer to: <http://en.wikipedia.org/wiki/Trapezoid>

² For more information, refer to: http://en.wikipedia.org/wiki/Gnome_sort

- Write a value returning MIPS function, *findAverage()*, to find the floating point average of an array. The function must call the *findSum()* function and perform the required type conversions.
- Write a value returning MIPS assembly language function, *iSqrt()*, to compute the integer square root of an integer. To estimate the square root of a number, set the initial estimate, *iSqrt_{est}* to the number and use the following algorithm iterated 50 times.

$$iSqrt_{est} = \frac{\left(\frac{iNumber}{iSqrt_{est}} \right) + iSqrt_{est}}{2}$$

- Write a void MIPS function, *displayStats()*, to print the diagonals array (seven per line) and the statistical information (minimum, maximum, median, and float average). Most of the code is provided, except for the loop to display the diagonals array. Five (5) spaces should be printed before each number and only seven numbers should be printed on each line in the format shown in the example.

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.
- 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):

```
MIPS Assignment #3
Trapezoid Diagonals Program

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

Diagonals - Values:
    150    154    154    155    156    158    159
    159    160    161    162    162    162    163
    166

Diagonals - Stats:
    min = 150
    max = 166
    med = 159
    float ave = 158.73333740

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

    . . .    output truncated    . . .
```