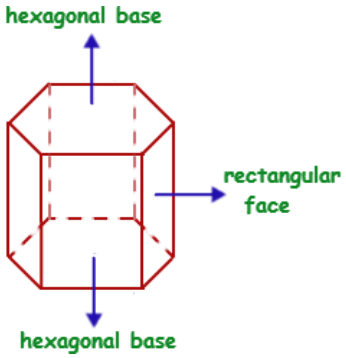
CS 218 – MIPS Assignment #2

Purpose: Become familiar with RISC Architecture concepts, the MIPS Architecture, and SPIM (the MIPS simulator).
Points: 50

Assignment:
Write a MIPS assembly language program to calculate the volume of each three dimensional hexagonal prism[1] in a series of hexagonal prisms.



The volume of a hexagonal prism is computed as follows:

$$hexVolume[i] = (3 * apothems[i] * bases[i] * heights[i])$$

Once the volumes are computed, the program should find the minimum, maximum, estimated median value, and average for the computed total volumes.

Since the list is not sorted, we will estimate the median value. Since the list length is even, the estimated median will be computed by summing the first, last, and two (2) middle values and then dividing by 4.

The program should display the results to the console window. The output should look something like the following (with the correct answers displayed and 4 numbers per line):

```
MIPS Assignment #2
  Hexagonal Volumes Program:
  Also finds minimum, middle value, maximum,
  sum, and average for the volumes.

  8996130  8343432  10642905  16329852
  12316920  12969000  10735452  15979680
  14863842  20054016  5960034  15707439
  [ ... truncated for space ... ]

Hexagon Volumes Minimum = ?
Hexagon Volumes Est Med = ?
Hexagon Volumes Maximum = ?
Hexagon Volumes Sum     = ?
Hexagon Volumes Average = ?
```

1 For more information, refer to: https://en.wikipedia.org/wiki/Hexagonal_prism

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 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 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.

MIPS Assignment #2 – Data Declarations

Use the following data declarations:

```
apothems:       .word       110,   114,   113,   137,   154
                .word       131,   113,   120,   161,   136
                .word       114,   153,   144,   119,   142
                .word       127,   141,   153,   162,   110
                .word       119,   128,   114,   110,   115
                .word       115,   111,   122,   133,   170
                .word       115,   123,   115,   163,   126
                .word       124,   133,   110,   161,   115
                .word       114,   134,   113,   171,   181
                .word       138,   173,   129,   117,   193
                .word       125,   124,   113,   117,   123
                .word       134,   134,   156,   164,   142
                .word       206,   212,   112,   131,   246
                .word       150,   154,   178,   188,   192
                .word       182,   195,   117,   112,   127
                .word       117,   167,   179,   188,   194
                .word       134,   152,   174,   186,   197
                .word       104,   116,   112,   136,   153
                .word       132,   151,   136,   187,   190
                .word       120,   111,   123,   132,   145

bases:          .word       233,   214,   273,   231,   215
                .word       264,   273,   274,   223,   256
                .word       157,   187,   199,   111,   123
                .word       124,   125,   126,   175,   194
                .word       149,   126,   162,   131,   127
                .word       177,   199,   197,   175,   114
                .word       244,   252,   231,   242,   256
                .word       164,   141,   142,   173,   166
                .word       104,   146,   123,   156,   163
                .word       121,   118,   177,   143,   178
                .word       112,   111,   110,   135,   110
                .word       127,   144,   210,   172,   124
                .word       125,   116,   162,   128,   192
                .word       215,   224,   236,   275,   246
                .word       213,   223,   253,   267,   235
                .word       204,   229,   264,   267,   234
                .word       216,   213,   264,   253,   265
                .word       226,   212,   257,   267,   234
                .word       217,   214,   217,   225,   253
                .word       223,   273,   215,   206,   213

heights:        .word       117,   114,   115,   172,   124
                .word       125,   116,   162,   138,   192
                .word       111,   183,   133,   130,   127
                .word       111,   115,   158,   113,   115
                .word       117,   126,   116,   117,   227
                .word       177,   199,   177,   175,   114
                .word       194,   124,   112,   143,   176
                .word       134,   126,   132,   156,   163
                .word       124,   119,   122,   183,   110
                .word       191,   192,   129,   129,   122
                .word       135,   226,   162,   137,   127
                .word       127,   159,   177,   175,   144
                .word       179,   153,   136,   140,   235
                .word       112,   154,   128,   113,   132
                .word       161,   192,   151,   213,   126
                .word       169,   114,   122,   115,   131
                .word       194,   124,   114,   143,   176
                .word       134,   126,   122,   156,   163
                .word       149,   144,   114,   134,   167
                .word       143,   129,   161,   165,   136

hexVolumes:     .space      400

len:            .word       100

volMin:         .word       0
volEMid:        .word       0
volMax:         .word       0
volSum:         .word       0
volAve:         .word       0
```

Note, the **`.space 400`** directive reserves 400 bytes which will be used to store 100 words.