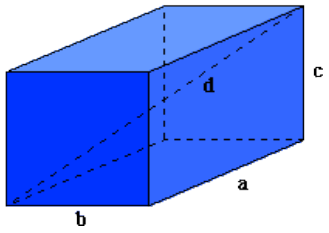
CS 218
Homework, MIPS Asst. #1

Purpose:	Become familiar with RISC Architecture concepts, the MIPS Architecture, and QtSpim (the MIPS simulator).
Due:	Friday (7/01)
Points:	45

Assignment:

Write a simple assembly language program to calculate the some geometric information for each rectangular parallelepiped in a series of rectangular parallelepipeds. Specifically, the program will find the volume for each of the rectangular parallelepipeds in a set of rectangular parallelepipeds. Once the values are computed, the program should find the minimum, maximum, middle value, and average for the volumes.



Note, for an odd number of items, the middle value is defined as the middle value. For an even number of values, it is the integer average of the two middle values. The data does ***not*** need to be sorted.

$$volumes[n] = aSides[n] * bSides[n] * cSides[n]$$

The program must display the results to the console window. The volumes should be displayed six (6) per line (they do not need to be justified). The output should look something like the following (with the correct answers displayed):

```
MIPS Assignment #1
  Program to calculate the volume of each rectangular
  parallelepiped in a series of rectangular parallelepipeds.
  Also finds min, med, max, sum, and average for volumes.

  Volumes:
    3863654     3079692     2186145     6427036     6940534     1522850
    1811460     2525627     5443181     6973830    -6375336     -3270046

     [...truncated for space...]

  Volumes Min = ?
  Volumes Med = ?
  Volumes Max = ?
  Volumes Sum = ?
  Volumes Ave = ?
```

Submission:

When complete, submit:

- A copy of the **source file** via the class web page by class time.
  Assignments received after the start time of class will not be accepted.

Provided Data:
Use the following data:

```
aSides:             .word       31,    21,    15,    28,    37
                    .word       10,    14,    13,    37,    54
                    .word      -31,   -13,   -20,   -61,   -36
                    .word       14,    53,    44,    19,    42
                    .word      -27,   -41,   -53,   -62,   -10
                    .word       19,    28,    24,    10,    15
                    .word      -15,   -11,   -22,   -33,   -70
                    .word       15,    23,    15,    63,    26
                    .word      -24,   -33,   -10,   -61,   -15
                    .word       14,    34,    13,    71,    81
                    .word      -38,    73,    29,    17,    93

bSides:             .word      101,   132,   111,   121,   142
                    .word      133,   114,   173,   131,   115
                    .word     -164,  -173,  -174,  -123,  -156
                    .word      144,   152,   131,   142,   156
                    .word     -115,  -124,  -136,  -175,  -146
                    .word      113,   123,   153,   167,   135
                    .word     -114,  -129,  -164,  -167,  -134
                    .word      116,   113,   164,   153,   165
                    .word     -126,  -112,  -157,  -167,  -134
                    .word      117,   114,   117,   125,   153
                    .word     -123,   173,   115,   106,   113

cSides:             .word     1234,  1111,  1313,  1897,  1321
                    .word     1145,  1135,  1123,  1123,  1123
                    .word    -1254, -1454, -1152, -1164, -1542
                    .word     1353,  1457,   182,  1142,  1354
                    .word    -1364, -1134, -1154, -1344, -1142
                    .word     1173,  1543,  1151,  1352,  1434
                    .word    -1355, -1037,  -123, -1024, -1453
                    .word     1134,  2134,  1156,  1134,  1142
                    .word    -1267, -1104, -1134, -1246, -1123
                    .word     1134,  1161,  1176,  1157,  1142
                    .word    -1153,  1193,  1184,  1142,  2034

volumes:  .space    220

len:                .word     55

vMin:               .word     0
vMid:               .word     0
vMax:               .word     0
vSum:               .word     0
vAve:               .word     0
```

Note, the **`.space 220`** directive reserves 220 bytes which will be used to store 55 4-byte words.