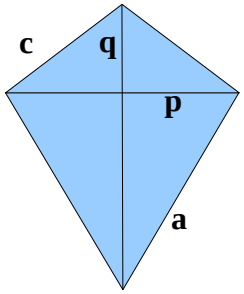
CS 218 – Assignment #5

Purpose: Learn to use arithmetic instructions, control instructions, compare instructions, and conditional jump instructions.
Points: 80

Assignment:

Write a simple assembly language program to calculate some geometric information for each kite (see diagram to the right) in a series of kites. Specifically, the program will find the area and perimeter for each of the kite in a set of rectangular kites. Once the values are computed, the program should find the minimum, maximum, middle value, sum, and average for the areas and perimeters.



$$kiteAreas[n] = \frac{pSides[n] * qSides[n]}{2}$$

$$kitePerims[n] = 2 \times aSides[n] \times cSides[n]$$

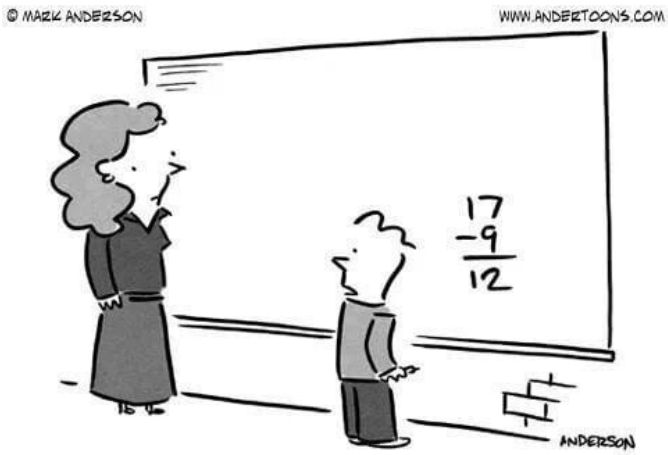
Since the list is not sorted, we will estimate the median value. Since the list length is odd, the estimated median will be computed by summing the first, last, and middle values and then dividing by 3.

Do ***not*** change the sizes/types of the provided data sets. All data is *unsigned*. As such, the DIV/MUL would be used (not IDIV/IMUL). Also, CDW/CWD/CDQ would **not** be used (as they are for signed data). The JA/JB/JAE/JBE must be used (as they are for unsigned data).

There is no provided main. Create the program source file based on the previous assignments. You may declare additional variables as needed.

Hint:

Pay close attention to the data types. The *pSides[]* array is word sized, the *qSides[]* array is double-word sized, the *aSides[]* array is byte sized, and the *cSides[]* array is doubleword sized.



"I know it's wrong, I'm just waiting for the autocorrect."

Submission:

- All source files must assemble and execute on Ubuntu with **`yasm`**.
- 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 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 loss of up to 10%.

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	10%	Must include header block in the required format (see above).
General Comments	20%	Must include an appropriate level of program documentation.
Program Functionality (and on-time)	70%	Program must meet the functional requirements as outlined in the assignment. Must be submitted on time for full score.

Assignment #5 Provided Data Set:

Use the following data declarations for assignment #5. *Note*, the assembler **is** case sensitive.

```
;  Provided Data
aSides        db        10,    14,    13,    37,    54
              db        31,    13,    20,    61,    36
              db        14,    53,    44,    19,    42
              db        27,    41,    53,    62,    10
              db        19,    28,    14,    10,    15
              db        15,    11,    22,    33,    70
              db        15,    23,    15,    63,    26
              db        24,    33,    10,    61,    15
              db        14,    34,    13,    71,    81
              db        38,    73,    29,    17
cSides        dd      1145,  1135,  1123,  1123,  1123
              dd      1254,  1454,  1152,  1164,  1542
              dd      1353,  1457,  1182,  1142,  1354
              dd      1364,  1134,  1154,  1344,  1142
              dd      1173,  1543,  1151,  1352,  1434
              dd      1355,  1037,  1123,  1024,  1453
              dd      1134,  2134,  1156,  1134,  1142
              dd      1267,  1104,  1134,  1246,  1123
              dd      1134,  1161,  1176,  1157,  1142
              dd      1153,  1193,  1184,  1142
pSides        dw       133,   114,   173,   131,   115
              dw       164,   173,   174,   123,   156
              dw       144,   152,   131,   142,   156
              dw       115,   124,   136,   175,   146
              dw       113,   123,   153,   167,   135
              dw       114,   129,   164,   167,   134
              dw       116,   113,   164,   153,   165
              dw       126,   112,   157,   167,   134
              dw       117,   114,   117,   125,   153
              dw       123,   173,   115,   106
qSides        dd      2183,  2372,  3231,  3121,  2153
              dd      3254,  1342,  5341,  4158,  1523
              dd      2125,  3133,  7384,  2274,  2114
              dd      5645,  1371,  3123,  3317,  1923
              dd      1634,  2334,  1156,  4164,  2742
              dd      3453,  4153,  2284,  2142,  3144
              dd      5345,  5130,  1423,  2113,  4123
              dd      2434,  1334,  3056,  3184,  1242
              dd      2353,  2153,  2284,  1142,  2334
              dd      3145,  1934,  2123,  4113
length        dd     49
aMin          dd     0
aeMed         dd     0
aMax          dd     0
aSum          dd     0
aAve          dd     0
pMin          dd     0
peMed         dd     0
pMax          dd     0
pSum          dd     0
pAve          dd     0

; Uninitialized data

section       .bss
kiteAreas     resd   49
kitePerims    resd   49
```

Note, the ".bss" section is for uninitialized data. The "resd" is for reserve doublewords.