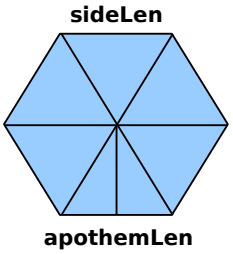
CS 218 – Assignment #5

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 hexagon[1] in a series of hexagons. Specifically, the program will find the area and perimeter for each hexagon in a set of hexagons. Once the values are computed, the program should find the minimum, maximum, estimated median value, sum, and average for the areas and perimeters.



Since the data is not sorted, we will obtain the estimated median value as follows. For an even length list, the estimated median value is computed by summing the two middle values and dividing by 2. For an odd length list it is just the middle value.

The formulas for hexagon area and perimeter are as follows:

$$hexPerims[i] = 6 * sideLens[i]$$

$$hexAreas[i] = \frac{hexPerims[i] * apothemLens[i]}{2}$$

The side length is the length of a singe hexagon side. The apothem is the distance between the middle of the side and the center of the hexagon.

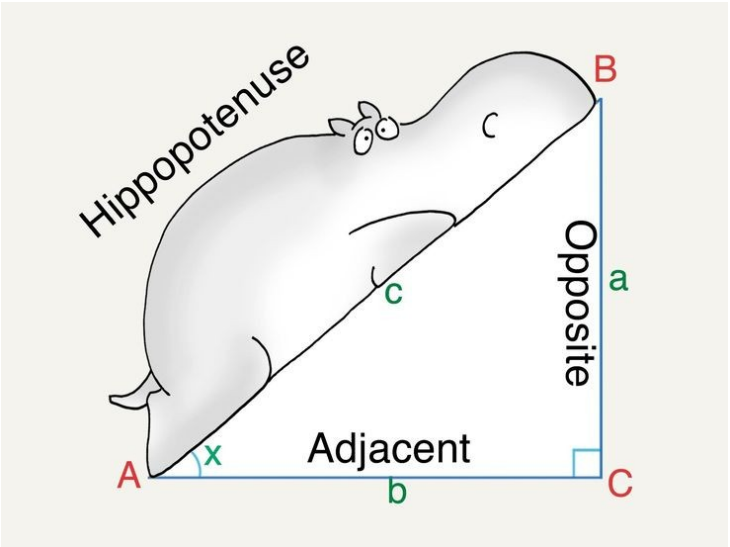
Do ***not*** change the sizes/types of the provided data sets. All data is *unsigned*. As such, the DIV/MUL would be used (not IDIV/IMUL). The JA/JB/JAE/JBE must be used (as they are for unsigned data).

There is no provided main.
You may declare additional variables as needed.

Hints:

Pay close attention to the data types. The *sideLens[]* array is double-word sized and the *apothemLens[]* array is word sized.

Consider completing the perimeters calculations before attempting the areas calculations.



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

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 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: <4-digit-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	5%	Must include header block in the required format (see above).
General Comments	10%	Must include an appropriate level of program documentation.
Program Functionality (and on-time)	85%	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.

```
section     .data
;  Provided Data

sideLens    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,  2034

apothemLens 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,   113

length      dd    50

perimMin    dd    0
perimEstMed dd    0
perimMax    dd    0
perimSum    dd    0
perimAve    dd    0

areaMin     dd    0
areaEstMed  dd    0
areaMax     dd    0
areaSum     dd    0
areaAve     dd    0

; ------------------------------------------------------------
; Uninitialized data

section     .bss
hexPerims   resd  50
hexAreas    resd  50
```

Note, the ".bss" section is for uninitialized data. The "resd" is for reserve double-words.