****Assignment 4 Week 2****

If you did not complete Week 1 you can start here:[HW4Week1.py](https://canvas.ucdavis.edu/courses/253898/files/4462562/download?verifier=oV9CKfPMH2SWKsYwq1kYGw5eZsPMSTED1WWN7CqH)

****Northern California Temperature Anomaly****

In week 2 you will have your program calculate a ****moving average**** of the temperature anomaly data for the temperature anomaly file you downloaded from Canvas (the latitude and longitude used in the query was for Sacramento) during the month of July. We'll let the user enter an integer k. Then for each year we calculate the average of the k years before, the year itself, and the k years after. that year. For instance, if the user picked k = 3, for the year 1936 we would average the temperatures for 1933, 1934, 1935, 1936, 1937, 1938 and 1939. Remember that to average seven numbers, we add them up and divide the sum by seven. This method is also sometimes called a ****sliding window****, where the “window” is the interval over which you are taking the average. Since your program lets the user choose k, you can experiment with averaging over different time intervals to see what happens.

****Requirements****

****Part 4****(35 points)

****Output the moving average****

With the parameter k that you obtained from the user in Week 1, calculate the moving average. The moving average should only be calculated for a year when there are k entries in the list before and after that year. So the moving average output will start k years after the beginning of the input file, and stop k years before the end of the input file.

Modify the loop you used to output data to a file to output the moving average instead. You can reuse the write commands, but you will want to implement the moving average using a nested loop. The outer loop will traverse the list for each year for which a moving average is calculated. The inner loop will average k years before, current year, and k years after to create the average.

Implementation hints:

* Use an index variable in both the inner and outer loops to access the list. This allows you to start and stop the list traversal at any point. If you are using a for loop you would use the range function to obtain the sequence.
* You can assume the years are contiguous. Finding the year corresponding to the smoothed temperature could be implemented by adding the right number to your list index. Alternatively,  you could look the year up if you loaded the years as strings in their own list.

Submit this program to Kodethon as ****HW4pt4.py****

****Example output for Part 4****

Here is an example of the tempAnomaly.txt file the program produces when the user chooses to average over 60 years; there are not too many rows since there are only 19 years in the middle of the century when there are 60 years before and 60 years after over which to average.

Year Value  
1940 -0.2331  
1941 -0.2169  
1942 -0.2150  
1943 -0.2228  
1944 -0.2107  
1945 -0.1796  
1946 -0.1667  
1947 -0.1582  
1948 -0.1585  
1949 -0.1492  
1950 -0.1711  
1951 -0.1688  
1952 -0.1490  
1953 -0.1556  
1954 -0.1548  
1955 -0.1580  
1956 -0.1420  
1957 -0.1101  
1958 -0.1017

****Part 5****(10 points)

****Getting input file from user****

Getting input filename from user. Change your code so that the first prompt asks the user for the name of the .csv input file. Your program prints the blue part.

Temperature anomaly filename:astrayData.csv

If the file cannot be opened, print the following message and simply exit the program. The filename should be whatever the user typed, not necessarily astrayData.csv.

The file astrayData.csv could not be opened.

Submit this program to Kodethon as ****HW4pt5.py****

****Part 6**** (5 points)

****Plotting the output****

Run the program with k=20 and make a nice plot in your favorite program (Excel, Sheets, Numbers, etc). Upload a screenshot or pdf of that part to Kodethon.

Here is an example. Yours should look different in some way. A number of these will be randomly selected for grading with the objective of determining if people are submitting in good faith.