Calculations

Problem 3

Based on the land average temperature column, calculate the monthly averages for each month for all years combined between 1900 and 2015. A total of twelve averages. One average per month.

Code Execution: **(screenshot of the code output)**

```
Question 3

Month      Avg Temp
-----      -----
January    2.815
February   3.331
March      5.396
April      8.537
May        11.395
June       13.540
July       14.449
August     13.958
September  12.167
October    9.527
November   6.223
December   3.812
```

Output:

Monthly Averages between 1900 -
2015: January: 2.30
February: 3.00
March: 4.97
April: 8.24
May: 11.13
June: 13.31
July: 14.29
August: 13.74
September: 11.71
October: 8.94
November: 5.75
December: 3.22

Purpose: To showcase the breakdown of calculations, for the monthly averages based on each month according to the land average temperature column across the years spanning from 1900 to 2015.	**Conflicts:** In question 3 the main conflict we encountered was figuring out a method to calculate the average using the land temperature data from 1900 to 2015. We also struggled with defining variables like 'MAX_LENGTH' and 'temp' as creating arrays of strings without specifying their size. Additionally, we found it tricky to deal with tasks, like file handling dividing by magic numbers such as 116, and using feof().	**Outputs/Analysis:** The output reveals the details, about how the monthly averages based on each month from the land average temperature fluctuated between 1900 to 2015. Upon analyzing the provided information it becomes evident that the output exhibits a pattern of alternating increases and decreases rather than remaining constant. However, there is a rise of 0.997 degrees, between 1900-2015, monthly land average temperature.	**How we would approach next time:** One way to improve the efficiency of the code is to reduce the reliance, on if statements. Utilizing loops (, for/while) and switch statements could prove advantageous.

C operations we used to answer question #3 (Explanation of code)

For this problem, the monthly averages were to be determined using the land average temperature columns for the years 1900 to 2015. The code starts off with opening and reading the file “GlobalTemperatures.csv” as previously stated, extracting the date (year-month) and the two land average temperatures from each line and storing it in an array. Processing from the years 1900 to 2015, the code then initializes the variables for each month of their respective years and simply calculates the average of the two temperatures from the data. In total there should be a total of twelve average monthly temperatures outputted for each year written in a text file, “AverageMonthlyTemperatures.txt”.

Variable Declarations

An array named "monthlyaverage" is created to hold the temperatures for each month of the year. Since there are 12 months in a year the array size is set to 12. The array "months" is defined as a two character array where each row stores the name of a month. The length of each month name is limited by the MAX_LENGTH defined elsewhere, in the code.

Reading and Processing Data from File

The file named "GlobalTemperatures.csv" is opened in read mode ("r"). A loop using do while is implemented to read each line from the file until reaching the end (feof(file)). For every line;fgets(temp, MAX_LENGTH, file); reads a line from the file into a buffer called temp. strtok(temp, " "); is applied to

elements in temp based on commas. The first strtok call retrieves the token (likely representing a date, in YYYY MM DD format). Isn't directly utilized. monthlyavg = strtok(NULL, " "); fetches the token likely representing the average temperature as a string.The program first breaks down the date token into parts using a hyphen to extract the year component ensuring that it is equal, to or greater than 1900.

When there is a temperature value, indicated by monthlyavg not being NULL it gets converted to a double. Then added to the sum of temperatures for that particular month in monthlyaverage[count]. The count variable increments through the months resetting back to 0 once it reaches 12.Certain variables such, as count, temp, token, monthlyavg, year, file and i need to be declared in the code. Additionally it assumes the presence of a called MAX_LENGTH which should be defined elsewhere in the code for it to compile and execute correctly.