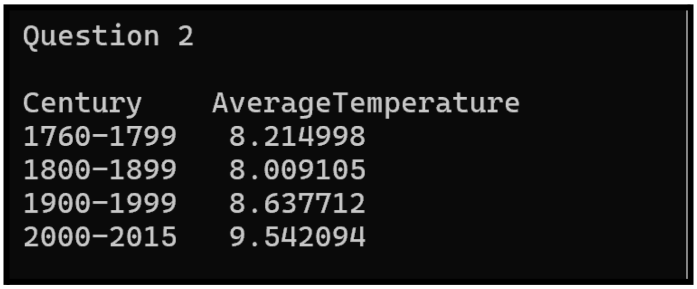
Calculations

Problem 2

Based on the land average temperature column, calculate the average land temperature for the different centuries: 18th century (1760-1799), 19th century (1800-1899), 20th century (1900-1999) and 21st century (2000-2015). One average per century.

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



```
Question 2

Century    AverageTemperature
1760-1799   8.214998
1800-1899   8.009105
1900-1999   8.637712
2000-2015   9.542094
```

Output:

18th century temperature average is 8.215

19th century temperature average is 8.009

20th century temperature average is 8.638

21st century temperature average is 9.542

Purpose:	**Conflicts:**	**Outputs/Analysis:**	**How we would approach next time:**
To demonstrate statistics that provide a deeper understanding of how the average temperature, in Celsius, has varied across various centuries between the 18th to 21st. This encompasses the impact on a level, during historical periods.	When it came to the challenge, merging questions 1 and 2 in a single file turned out to be more difficult than expected. Some other conflicts encountered during this process include setting up variables like "centuryavg[]". Yearaverage[]". Additionally the code tries to divide "century" by "count" without ensuring that "count" is not zero.	The output reveals the details, about how the average temperature in Celsius fluctuated between the century 18th to 21st centuries. Upon analyzing the provided information it becomes evident that the average land temperature for the different centuries exhibits a pattern of alternating increases and decreases rather than remaining constant. However, there is a rise of	A comparable method could be used next time through the use of effective techniques for determining the average land temperature over various centuries would be utilized preferably.

	This leads to a division by zero error if the first year checked ("i % 100 == 0") does not result in an increase of "count". Moreover managing types and declarations posed a challenge making sure all variables, like 'yearaverage' 'i' 'count' are properly declared with types such as int, float and double. These are some mistakes encountered along with others like century average calculation, for the first century.	2.64 degrees, between the 18th and 21st century is approximately 1.32.	

C operations we used to answer question #2 (Explanation of code)

Setting Up Initial Values;

We set count and n to 0. Count is used to track the number of years considered in calculating the temperature for each century while n serves as an index, for storing these calculated temperatures into the centuryavg array.
The line printf("\nQuestion 2\n\n"); simply displays a title. Heading for this section of output.
Displaying Column Headings;

Next it prints "Century AverageTemperature\n" to show column titles that will be used in the output indicating each century alongside its temperature.

Looping for Temperature Calculation;

A for loop runs from 1760, to 2016 (1760 + 256) computing the temperature for each century within this time frame.Within this loop if (i % 100 == 0 || i == 2016) checks if a year marks either a centurys end or reaches the data ranges endpoint (2016). It's worth mentioning that when a year is divisible, by 100 (like 1800 or 1900) it signals the conclusion of a century. Once we reach the end of a century (. In the case of 2016, which marks the end of our data range) we divide the accumulated temperature for that century by the count (representing the number of years within that century) to determine the average temperature for that period. The resulting average temperature is then saved in the array called centuryavg[n]. A printf statement is used to display both the range of years for that century (1760 1799) and its corresponding average temperature. Following this we reset the variables count and century in preparation for calculating averages for the century. Specifically n is incremented to move to the position in the centuryavg array count is reset to zero to start counting years in the century and century is reset to zero for accumulating temperature values anew. In each iteration, through our loop representing each year we add up each years temperature (stored as yearaverage[i 1760]) to our tally for that century under consideration. This process helps us sum up all temperatures across each year within a given century. Additionally incrementing count++ ensures we keep track of how years contribute towards calculating an average for our ongoing century.

Initialization steps are taken into account for Monthly Averages;

After computing century averages the program sets up a monthlyaverage array, with 12 slots to hold the temperatures for each month. However it doesn't actually use this array for any calculations in the code provided. Additionally it creates a months array containing the names of all twelve months for linking temperatures to specific months, in later sections of the program.