

COS 135 Individual Assignment #8

Due: Friday 04/10/20 End of the day

This assignment has 4 pages. Please submit a .zip file with completed source codes and output files for each problem.

(100pts) write separate C programs for following tasks and submit your source codes. DO NOT MODIFY THE GIVEN DATASETS. C program templates are also provided for your convenience.

Special instructions:

Comments are required in the following locations:

- At the top of the source code comment your name and a short program description.
- Comment the purpose of variables, functions, and other elements in your code.
- Comment major sections of code such as input, processing, and output.

Program Design:

Your program is a professional document and must be neat and easy to read. All programs should follow the listed specifications.

- Comments should be aligned and entered in a consistent fashion
- Blank lines should be added to aid readability
- Code within blocks should be indented
- Comments should not contain spelling mistakes
- Variable names should be meaningful
- **Define functions and data structures where necessary**
- **Optimize your code: least possible number of lines to produce the output**
- **Error handling: you should handle all the possible error conditions and invalid inputs**

A). (40pts) File **aircraft_arrivals_departures.csv** contains monthly aircraft arrival and departure data recorded at Bangor airport from 2010 to 2015. You may use Linux *cat* command or *VIM editor* to print its content on the terminal to see comma separated values (.csv) and its format. More information on .csv format: https://en.wikipedia.org/wiki/Comma-separated_values

Your tasks:

1. **Complete the given C program (assignment8a-template.c)** to analyze the data using multidimensional arrays, and output average number of arrivals and departures for each year – in the following format.

=====

Yearly average arrivals and departures:

Year	Arrivals	Departures
2010	10980.75	10985.33
2011	12568.92	12573.67
2012	13529.08	13531.08
2013	14320.83	14326.25
2014	14223.33	14225.50
2015	14429.33	14431.83

=====

2. As provided in the template (assignment8a-template.c) program, separate functions should be defined for: a) reading data, b) calculate yearly average values, and c) print data.

Following are the important steps:

1. Load data from the .csv file to a multidimensional array (you need to parse each line from the .csv file and save data in a multidimensional array) – COMPLETED!
2. Develop an algorithm to calculate average values for each year – **FOR YOU TO COMPLETE!**
3. You may save average values in the second array – **FOR YOU TO COMPLETE!**
4. Output yearly average values (with two decimal points) – COMPLETED!

Sample output:

```
nimesha@nimesha-VirtualBox:/media/sf_shared/assignment-week-10$ gcc assignment8a-complete.c -o a
nimesha@nimesha-VirtualBox:/media/sf_shared/assignment-week-10$ ./a
Year    Arrivals    Departures
2010    10980.75    10985.33
2011    12568.92    12573.67
2012    13529.08    13531.08
2013    14320.83    14326.25
2014    14223.33    14225.50
2015    14429.33    14431.83
nimesha@nimesha-VirtualBox:/media/sf_shared/assignment-week-10$
```

B). (60pts) Consider the dataset provided: **sunspots_dataset.csv**

Context: This given univariate dataset contains all the occurrences of sunspots (approximate numbers yearly from 1700 to 1989).

Write a C program to:

1. Read the dataset from the .csv file to an appropriate multidimensional array.
2. Classify the dataset decennially (i.e. calculate average for every ten years, for example, 1700 – 1709, then 1710 – 1719, until 1980 - 1989).

Use another multidimensional array to store the classified data (decennially).

3. Then, save classified data in a .csv file: **sunspots_dataset_10.csv**

```
decennially, average_sunspots, normalized
1700, 216.0, 2
1710, 252.0, 2
1720, 524.0, 5
.
.
1980, 783.4, 7
```

Use the following method to calculate the normalized values:

$$\text{normalized_value} = \text{floor}(\text{average_sunspots_for_a_decade} / 100)$$

(**Tip:** you need to include math.h and compile your code with -lm switch)

Note: sunspots_dataset_10.csv is provided for you to test your program's output. Average sunspot values should only contain one decimal point.

4. Finally, outputs a simple visualization based on the normalized average values as shown in next page.
5. Define functions where necessary.

```
nimesha@nimesha-VirtualBox:/media/sf_shared/assignment-week-10$ gcc assignment8b.c -o a -lm
nimesha@nimesha-VirtualBox:/media/sf_shared/assignment-week-10$ ./a
1700, 216.0, 2 * *
1710, 252.0, 2 * *
1720, 524.0, 5 * * * * *
1730, 511.0, 5 * * * * *
1740, 367.9, 3 * * *
1750, 375.6, 3 * * *
1760, 537.5, 5 * * * * *
1770, 713.9, 7 * * * * * * *
1780, 712.4, 7 * * * * * * *
1790, 359.0, 3 * * *
1800, 275.1, 2 * *
1810, 208.8, 2 * *
1820, 270.2, 2 * *
1830, 673.5, 6 * * * * * *
1840, 572.3, 5 * * * * *
1850, 427.1, 4 * * * *
1860, 488.8, 4 * * * *
1870, 512.8, 5 * * * * *
1880, 377.3, 3 * * *
1890, 449.6, 4 * * * *
1900, 355.3, 3 * * *
1910, 391.5, 3 * * *
1920, 420.3, 4 * * * *
1930, 511.0, 5 * * * * *
1940, 720.2, 7 * * * * * * *
1950, 916.8, 9 * * * * * * * *
1960, 609.1, 6 * * * * * *
1970, 616.0, 6 * * * * * *
1980, 783.4, 7 * * * * * * *
nimesha@nimesha-VirtualBox:/media/sf_shared/assignment-week-10$
```