I210: Information Infrastructure I – Mastery Check 5

When you have completed this Mastery Check, you should:

1. ***Submit your solution to Canvas under "Mastery Check 5"***
2. ***Upload a copy of your work to OneDrive as a backup in case something goes wrong with Canvas.***

Make sure you click through to the final "Submit" button on Canvas. You are responsible for making sure your work is submitted **by the end of the lab**, so it is strongly advised that you verify the submission before leaving. **We CANNOT accept late submissions.**

As a reminder, you may use anything on the Canvas section for **I210 only**, as well as the course textbook. You may use any notes (physical or online) taken for **I210 only**. You may work on an STC machine or on your personal laptop.

You may NOT use the Internet *except*:

- the Canvas sections for I210
- any group coding or note spaces you've set up for I210.

Using code you found online from outside of this class or code that you or your group did not write that is not from the book or slides is likely to constitute academic misconduct.

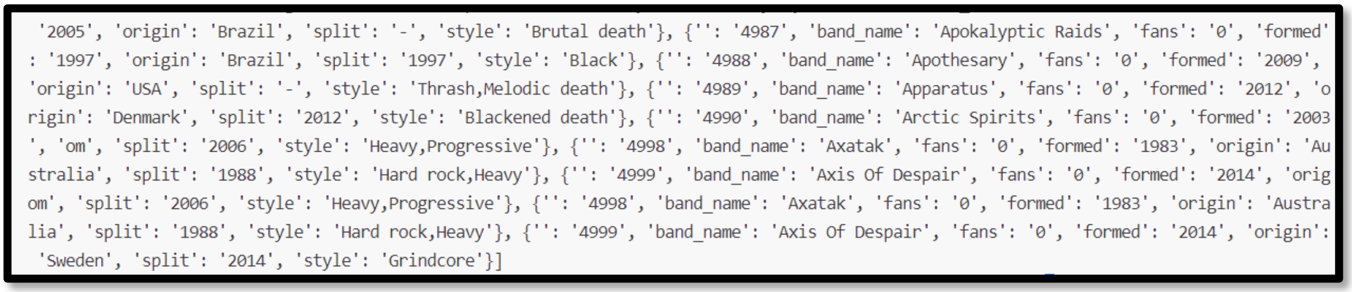
INFO I210 Mastery Check 5: Data is So Metal, Band Analysis

Objective:
To develop a Python program that reads a csv file then analyzes and filers the data in a meaningful way. Through this project, you will gain hands-on experience with concepts including file handling, dictionaries, loops, and functions.

Project Description:
The project will be divided into several stages, each designed to reinforce key programming concepts while contributing to the overall goal of creating an analysis tool. **Be sure to add comments throughout your code.**

Part 1: Read data from the csv file

- Define and complete a function named **read_data(file_name)** that takes a string as a file name, opens the file, and reads the contents of the file using the csv module. We suggest using the DictReader to read the data.
- Create a nested list to store the data, where each element of the list is a row of data. Depending on which method you use to read the data, each row will either be represented as a list or a dictionary.
- Return the nested list
- If you were to test this function and print the returned data, it could look something like this:



```
 '2005', 'origin': 'Brazil', 'split': '-', 'style': 'Brutal death'}, {'': '4987', 'band_name': 'Apokalyptic Raids', 'fans': '0', 'formed'
: '1997', 'origin': 'Brazil', 'split': '1997', 'style': 'Black'}, {'': '4988', 'band_name': 'Apothesary', 'fans': '0', 'formed': '2009',
'origin': 'USA', 'split': '-', 'style': 'Thrash,Melodic death'}, {'': '4989', 'band_name': 'Apparatus', 'fans': '0', 'formed': '2012', 'o
rigin': 'Denmark', 'split': '2012', 'style': 'Blackened death'}, {'': '4990', 'band_name': 'Arctic Spirits', 'fans': '0', 'formed': '2003
', 'om', 'split': '2006', 'style': 'Heavy,Progressive'}, {'': '4998', 'band_name': 'Axatak', 'fans': '0', 'formed': '1983', 'origin': 'Au
stralia', 'split': '1988', 'style': 'Hard rock,Heavy'}, {'': '4999', 'band_name': 'Axis Of Despair', 'fans': '0', 'formed': '2014', 'orig
om', 'split': '2006', 'style': 'Heavy,Progressive'}, {'': '4998', 'band_name': 'Axatak', 'fans': '0', 'formed': '1983', 'origin': 'Austra
lia', 'split': '1988', 'style': 'Hard rock,Heavy'}, {'': '4999', 'band_name': 'Axis Of Despair', 'fans': '0', 'formed': '2014', 'origin':
 'Sweden', 'split': '2014', 'style': 'Grindcore'}]
```

*Image is cropped. This is the end of the data output

Take Note: You'll be using the read_data() function as the first step in each of your other functions as we define them below.

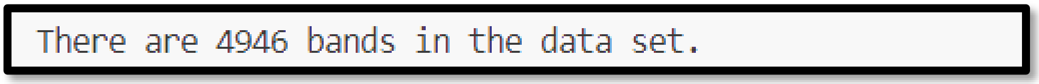
When you call read_data(), you need to pass in the file name. It is your choice how you do that. You can utilize a user input to get the file name and pass that into each new function so that you can pass it to read_data() or you can hard code the file name into each new function when you call read_data(). Please do not hard code the file name into read_data()

Part 2: Counting and filtering bands

Let's use this data to answer a few questions through code. We'll start simply.

First, how many bands are represented in our data set?

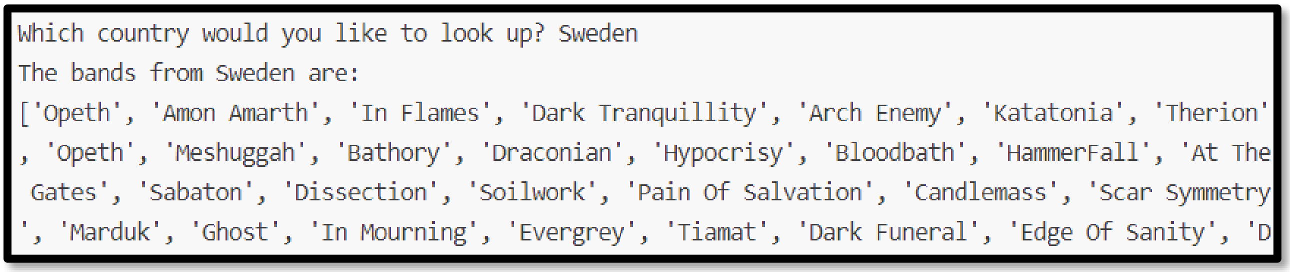
- Define a **num_total_bands()** function that first reads the data by calling your **read_data()** function.
- Calculates how many bands are in the data set.
- Return an integer representing the number of bands.
- Call your function to display the total number of bands in the data set. It may look something like this:



```
There are 4946 bands in the data set.
```

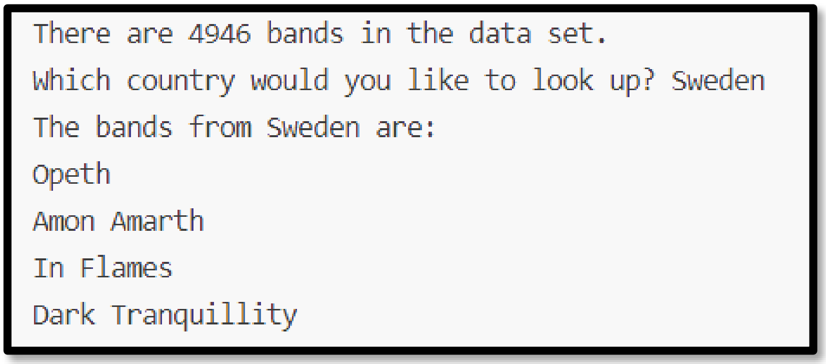
Next, let's filter the data to answer some additional questions.

- Define a function called **get_bands_from_country(country)** that takes the name of a country as input and returns a list of the names of all the bands that formed in that country.
    - Call your function to display the bands from that country in the data set. It may look something like this:



```
Which country would you like to look up? Sweden
The bands from Sweden are:
['Opeth', 'Amon Amarth', 'In Flames', 'Dark Tranquillity', 'Arch Enemy', 'Katatonia', 'Therion'
, 'Opeth', 'Meshuggah', 'Bathory', 'Draconian', 'Hypocrisy', 'Bloodbath', 'HammerFall', 'At The
 Gates', 'Sabaton', 'Dissection', 'Soilwork', 'Pain Of Salvation', 'Candlemass', 'Scar Symmetry
', 'Marduk', 'Ghost', 'In Mourning', 'Evergrey', 'Tiamat', 'Dark Funeral', 'Edge Of Sanity', 'D
```

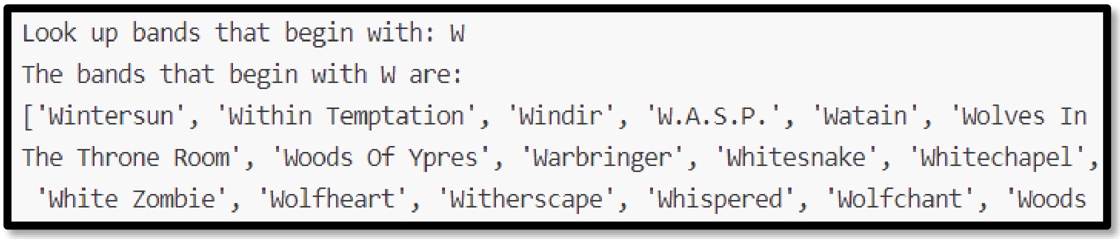
    - Optional bonus: Have the output look better than simply displaying the list. One option may look something like this:



```
There are 4946 bands in the data set.
Which country would you like to look up? Sweden
The bands from Sweden are:
Opeth
Amon Amarth
In Flames
Dark Tranquillity
```

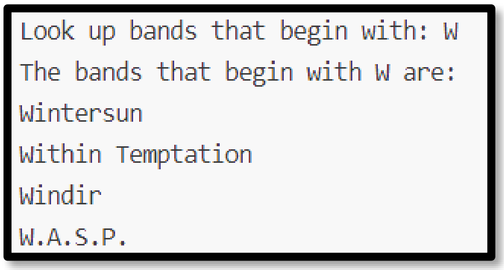
*cropped image

- Define a function called **get_bands_starting_with(letter)** that takes a letter as input and returns a list of the names of bands whose name begins with that letter.
    - Call your function to display the bands with names beginning with that letter in the data set. It may look something like this:



```
Look up bands that begin with: W
The bands that begin with W are:
['Wintersun', 'Within Temptation', 'Windir', 'W.A.S.P.', 'Watain', 'Wolves In
The Throne Room', 'Woods Of Ypres', 'Warbringer', 'Whitesnake', 'Whitechapel',
 'White Zombie', 'Wolfheart', 'Witherscape', 'Whispered', 'Wolfchant', 'Woods
```

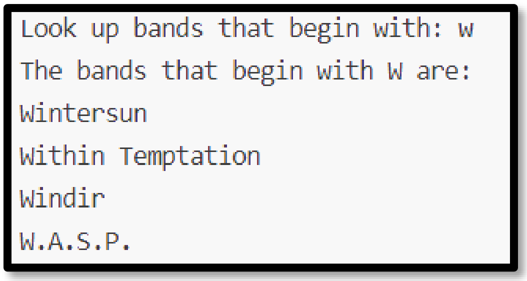
    - Optional bonus: Have the output look better than simply displaying the list. One option may look something like this:



```
Look up bands that begin with: W
The bands that begin with W are:
Wintersun
Within Temptation
Windir
W.A.S.P.
```

*cropped image

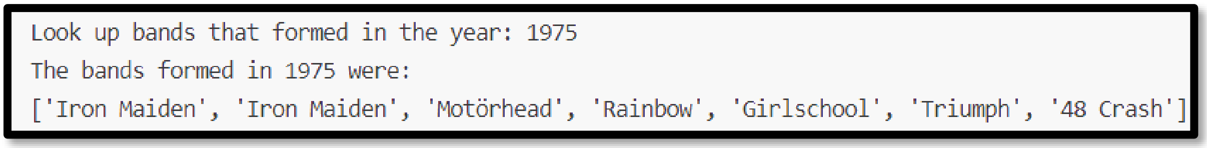
    - Optional bonus: To avoid returning an empty list, allow for the input letter to be either lower case or upper case (W vs w)



```
Look up bands that begin with: w
The bands that begin with W are:
Wintersun
Within Temptation
Windir
W.A.S.P.
```

*cropped image

- Define a function called **get_bands_formed_in_year(year)** that takes a year as input and returns a list of the names of bands who formed in that year.
    - Call your function to display the bands formed in the given year in the data set. It may look something like this:



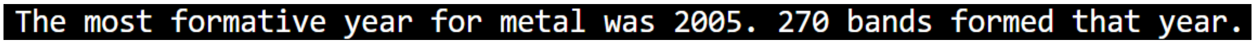
```
Look up bands that formed in the year: 1975
The bands formed in 1975 were:
['Iron Maiden', 'Iron Maiden', 'Motörhead', 'Rainbow', 'Girlschool', 'Triumph', '48 Crash']
```

    - Optional bonus: Have the output look better than simply displaying the list.

Part 3: Analysis

Finally, let’s do a bit more analysis to answer one last question: what was the most formative year in the history of metal? What was the year that the most bands formed?

- Define a function called **get_year_with_most_bands()** that tallies up the number of bands that were formed each year in our data set, calculates the year with the most number of bands, and returns the year and the number of bands formed that year.
- We suggest that you use a dictionary to tally up the number of bands formed each year, create a list from the dictionary, and sort the list by the number of bands.
- Call your **get_year_with_most_bands()** function and print out the year the most bands formed, as well as how many bands formed that year. It may look something like this:



```
The most formative year for metal was 2005. 270 bands formed that year.
```

- Optional bonus: List the bands formed in the ‘most formative year’ sorted alphabetically A-Z

Part Bonus 1:

Are there any other simple questions you can answer through code? Perhaps you were wondering which country has produced the most bands, or which bands play a given style. Write a function to help you answer this question and call that function in the main part of the program. Display the result in a way that explains your new insight to the user. Include comments in your code to explain the code to your grader.

Part Bonus 2:

Are there places where it would be beneficial to build in error handling? Utilize try & except somewhere useful within your code. Does it function as you would anticipate? Include comments in your code to explain the code and its function to your grader.

Reminder: Maximum Score: 110pts