Report 08

1. **Reflections**

* What was the easiest and hardest part of this assignment?

Easiest part:

The average function is an easy part.

As long as you lock down the location of useful data in the csv file, it is not hard to find the way to make it a list.

Hardest part:

There are too many places that could possibly cause an error, it is hard to find them all, such as what if the user inputs a location or starting year which do not exist. You need to find them all and raise an error and give a message accordingly, protect the main function by try…except syntax.

A lot of details such as the transform of year and index of the list, skip the first line of the csv, 3 digits remain after dot, treatment of raw data to delete irrelevant part and get the useful information.

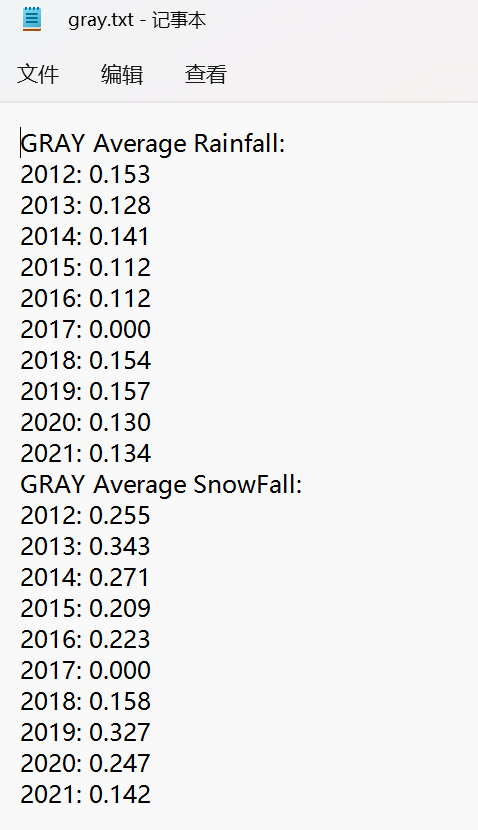
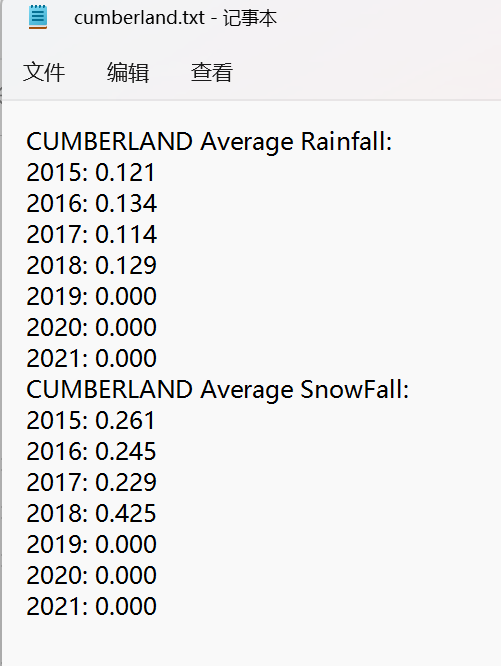
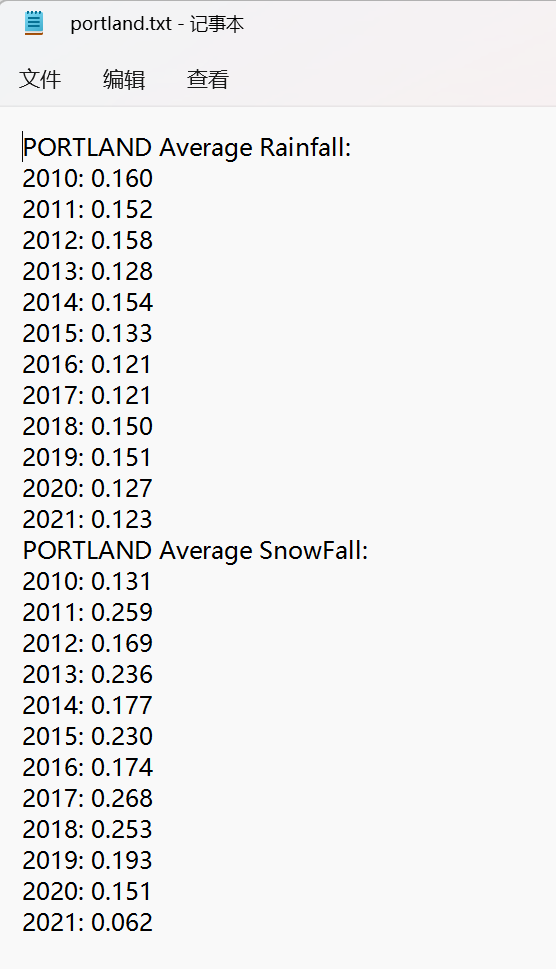
* What did you learn?

Raise an error in different condition and provide useful information based on the condition, handle errors by try.. except syntax.

A lot of practice on string and list methods to get useful information from raw data.

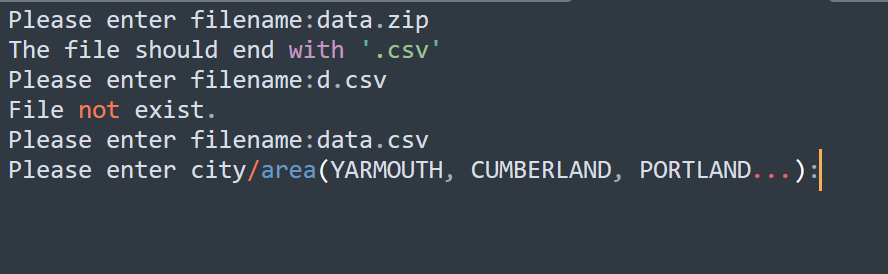
File I/O, read data in a csv file and write to an out put file.

1. **Output**

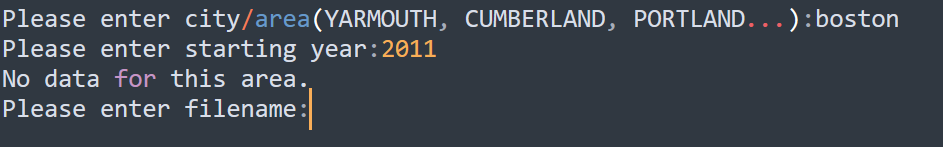


Output(Portland 2010~2021, Cumberland 2015~2021, yarmouth 2011~2021,gray2012~2021)

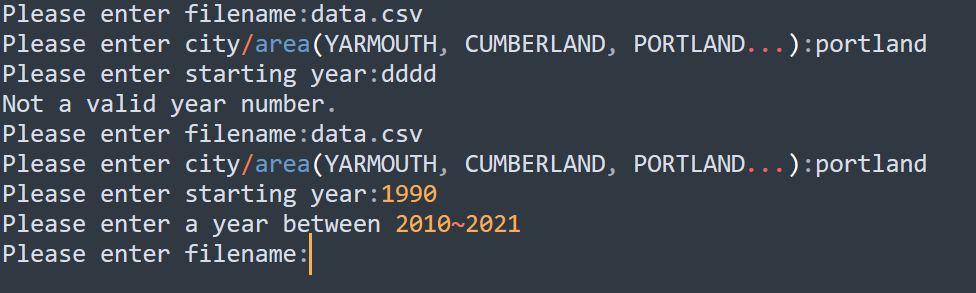
Filename error handling



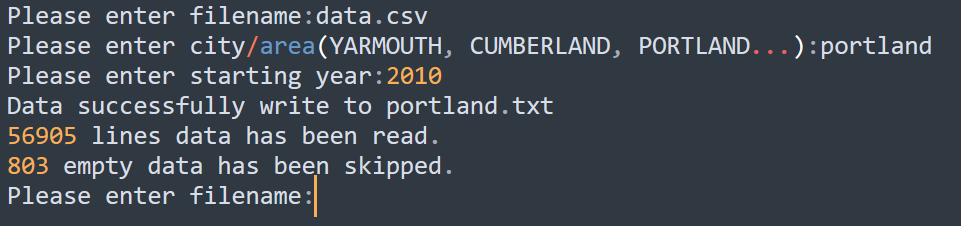
City name error handling



Starting year error handling



Complete operation of data extract



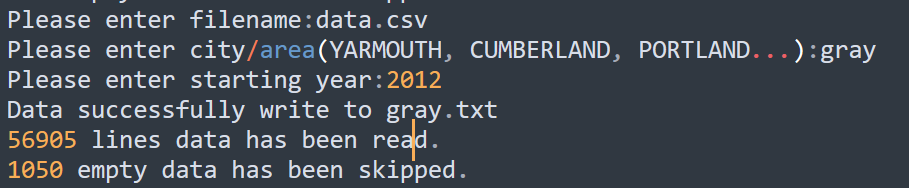
1. **Extensions**

* **Multiple cities instead of just Portland**

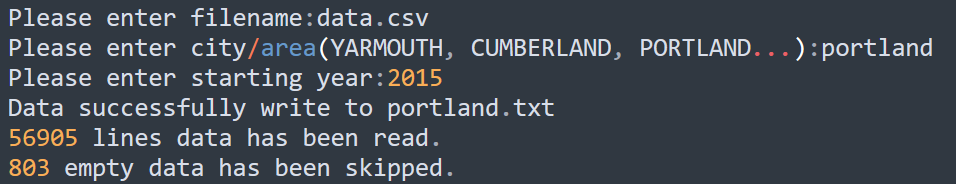
As can be seen in part 2, multiple cities in the csv file can be selected to get data.

* **Let user select city and starting year, gave a notification if no data for input city, negelect upper or lowercase input.**

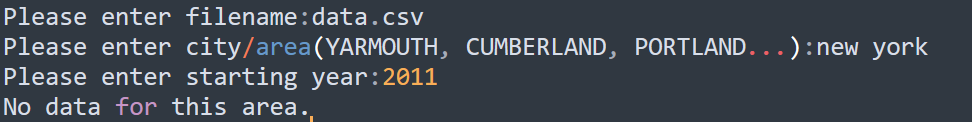
Filter data of certain city and starting year based on the user input, if no data found, the program will notify user “No data for this area”



Get data of gray start from 2021

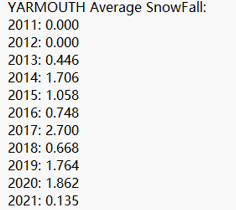


Get data of portland start from 2015



Data not found in new york

* **Snow data is Extracted.**



* **Add some other exceptions.**

Starting\_year value checking, raise a value error if value is not between 2010 and 2021

if starting\_year > 2021 or starting\_year < 2010:

**raise ValueError("Please enter a year between 2010~2021")**

Location checking, if user input location is not found, raise a value error.

if location\_found:

print\_avgs(rain\_data, snow\_data, city, starting\_year, f"{city}.txt")

print (f"{line\_num} lines data has been read.")

print (f"{empty\_data} empty data has been skipped.")

else:

**raise ValueError("No data for this area.")**

* **Line read and empty data counting.**

After extracting data, give number of lines read and skipped empty data.

