Second Homework Assignment

Due Date: **5/1/2024 at 5:00pm (25% of total grade)**

Problem 1 (40 points)

Data: /home/public/google

Code: /home/public/google/src

Description: The data for this problem comes from Google's project for counting word frequencies in its entire Google Books collection. You are given two files: one file reports 1 grams (single words) and another file reports 2 grams (pairs of words; one following each other in text). The data represents the number of occurrences of a particular word (or pairs of words) in a given year across all books available by Google Books. The number of volumes/books containing a word (or pairs of words) is also reported.

Write a MapReduce program in python that reports the average number of volumes per year for words containing
the following three substrings: 'nu,' 'chi,' 'haw'.

Example:

The 1 gram file format --the regex "\\s+" will match any kind of whitespace (space, tab etc):

> **word** \\s+ **year** \\s+ **number of occurrences** \\s+ **number of volumes** \\s+ …

The 2 gram file format:

> **word** \\s+ **word** \\s+ **year** \\s+ **number of occurrences** \\s+ **number of volumes** \\s+...

The final output should show the **year**, **substring**, and **average number of volumes** where the substring appears in that year. For example:

```
2000,nu,345

2010,nu,200

1998,chi,31
```

If each word in the bi-gram includes the string, it should be counted twice in the average. For example, for the bi-gram "nugi hinunu" with volume of 10, when calculating the average, its contribution to the numerator should be 2 times 10 and in the denominator it should be 2. A unigram counts only once regardless of the number of occurrences of "nu" in the word.

The ‘year’ column may include erroneous values which can be a string. If the year field is a string, the record should be discarded.

Folder /home/public/google/src contains a java implementation of this task. Use chatGPT as much as possible to create a python code to solving the same problem in map reduce.
Run your python code and make all of the necessary changes so that it works correctly.

On github submit the following:

1. chatGPT prompts
2. the python code
3. sample results

Problem 2 (60 points)

Data: /home/public/music

Description: The data for this problem is a subset of the million song database (https://www.kaggle.com/c/msdchallenge#description), and its full size is 42GB. The file is in csv format.

For each artist in the data set, compute the maximum duration across all of their songs. The output should be: **artist, max duration**.

The file is in csv format.
Song title (column 1), artist’s name (column 3) and duration (column 4)

Write your own python implementation of map reduce for solving this problem. You have to implement map, shuffle, and reduce. The map and reduce tasks must run in parallel by using the processing python module.

Split the input file to 20 smaller files (either by using linux commands or python). Your code should take the number of map processes and the number of reduce processes as input and then it should create the corresponding processes. Example: python songs.py 20 5 corresponds to 20 splits (map processes) and 5 reduce processes. You have flexibility how shuffle is implemented.

On github submit the following:

1. Readme.md that includes pseudo code behind your implementation
2. The python code
3. Sample results