

ALEX MALAVE

Assignment: Homework #5

Problem 1:

printvertical.py

My solution in *printvertical.py* contains one function, **main**. The function uses input in order to obtain a string and assign it to the variable *sent*. *sent* is then made into a list by using the *.split()* operation, and subsequently assigned to the variable *wordSplit*. The variables *tab* and *largestWord* are predefined as an empty list and an empty string, respectively.

First, a for loop checks for the largest word in the inputted sentence by comparing the lengths of the string values within the list *wordSplit*. Afterwards, another for loop goes through each of the words within *wordSplit*, and uses *largestWord*'s length in order to get spaces into the list objects that will go within *tab*. If the length of *tab* is 0 (it's empty), or if the index doesn't exist yet, but should, an entry is added with a new index into *tab*. The entry is determined to be either a space character or the next character in a given word within *wordSplit* depending on the size of the current word and the current index. Otherwise, if the index exists within *tab*, and the current word is within the bounds of the length of *largestWord*, it adds the next character of the next word to that index of *tab*, or it adds a space character.

Finally, the function runs a for loop to print out the formatted contents of *tab*. The code should be run as follows: *python3 printvertical.py*

Problem 2:

rating.py

My solution in *rating.py* contains three functions, **main**, **processColumnFormat**, and **processRowFormat**. Main handles the opening of both the input and output files, it discerns whether to use **processColumnFormat** or **processRowFormat** depending on the first line of the input file provided (which it does by using *.readline()*), and it handles the closing of both files once everything is finished.

processRowFormat takes two arguments, the input and output file objects, designated *iFile* and *oFile*. The function then uses an empty list *tab*, which it fills using a for loop. The function reads each line individually, while splitting each line into a list *s* using the *.split()* operation. It then checks if the line isn't empty by running *len(s) > 1* (a safety measure), and continues to assume that the name of the movie is the first index of the newly created list *s*, while gathering the other indexes (assumed to be the movie ratings) into a variable named *acc*. The function then appends the movie name with the calculated average of the scores for each movie to *tab*. Afterwards, the function uses another for loop to write each object in *tab* (which should be each movie's title and average rating) to the output file, *oFile*.

processColumnFormat takes two arguments, the input and output file objects, also designated *iFile* and *oFile*. The function then uses two empty dictionaries *numReviews* and *reviewSums*. Similar to **processRowFormat**, the function reads each line individually using a for loop, and assigns a variable *s* to a list created out of each line's contents by using the *.split()* operation. It also checks if the line isn't empty by running *len(s) > 1*, and then checks if the movie title already is an index within the dictionary

numReviews. If so, it adds one to the value of that index, and adds whatever the value is of the review (assumed to be the second index of the list *s*) to the dictionary *reviewSums* (using the movie title as the index). Otherwise, it creates a new index within *numReviews* and within *reviewSums*. After that has been done, a new list *tab* is created and formatted in the preferred way to see movie ratings (movie name, :, average score) through a for loop that goes through each index and value within *reviewSums*. Afterwards, the function uses another for loop to write each object in *tab* to the output file, *oFile*.

The code should be run as follows: `python3 rating.py <name of input file> <name of output file>`