

Homework 2 Descriptions

1. Dictionary.txt
 - a. For part a, I used a regex and grep function to see which words have 3 a's and I also checked if it starts with an a just to be safe even though I used -i flag which should ignore cases. At the end, I use wc to count the number of lines which also means to count the number of words in our case.
 - b. Part B, I created a regex which checks if a word has exactly 3 e's by making sure that the pattern ends after the last part where I check the remaining characters are not e's. After that I simply counted the number words using wc function.
 - c. First, I check for each word that has consecutive e's while also checking if it starts with an e and then I reverse all the words to extract the first 3 characters which are actually the last 3 and then I reverse it again to only keep the last 3 characters of each word. After this, I sort all of them so I can effectively use the uniq function after which I sort them again but now using the count of each last 3 characters. In the end, I use the head function to only keep the top 3.
2. I read in each line of the input file one after the other and for each line if it contains scanf or printf. If it does, I keep track of the counts of each along with the total number of lines outside of the while loop. In the end, I simply print out each statistic with the percentage of each with comparison to the total number of lines in the file.
3. For this problem, I traverse through the input directory and each one of its subdirectory and if I ever encounter a file in my function, I loop through each line of the file and check for valid email addresses using grep and regex. If it is valid email, I add it to the output

file and then once I am done going through each file in the given directory, I sort the output file while only keeping the unique email addresses using the sort function and its flags.

4. For this problem, I go through each line in the input file one by one and first check the city using the cut function. Then I check if the extracted city has 2 words using regex and if it is I extract the year from the line using regex and since grep only checks the first match in the line, the format works to my favor and effectively grabs the year. I then calculate the age using 2024 as the current year and then add it to a temporary 'age.txt' file. Lastly, I sort the temporary file using the age first and then the name and display it as standard output and then delete the temporary file.