Homework - 2

Problem - 1: (2 points)
Write a Bash script that:

1. Searches for all lines containing the word “scanf” or “printf” in a given ‘C’ file passed as an argument (using grep).
2. Counts the number of lines containing “scanf” and “printf” separately.
3. Redirect the lines containing “scanf” to a new file named scanf_log.txt and the lines containing “printf” to a new file named printf_log.txt.
4. If the file scanf_log.txt or printf_log.txt already exists, the script should append the lines to the existing file instead of overwriting it.
5. The script should also print the total number of lines in the input file and the percentage of lines containing “echo” and “printf”.

Problem - 2: (4 points)
Write a Bash script such that:

1. The script takes a single argument, which is the name of a directory. This directory should contain text files, each containing a list of email addresses. The email addresses can be spread across multiple lines within each file.
2. the script should recursively traverse the directory and its subdirectories to find all text files. (Do not use find method)
3. For each text file, the script should extract all unique email addresses contained within it. Use regular expressions to identify valid email addresses. Valid email addresses should follow the format "username@domain.com".
4. After extracting the email addresses from all the files, the script should combine them into a single list and remove any duplicates.
5. Finally, sort the list of unique email addresses in alphabetical order and save them to a new text file named "unique_emails.txt" in the same directory where the script is located.

Problem - 3: (4 points)
Write a Bash script where:

1. The script should first read a file line by line, where each line is a person's information formatted as "FirstName LastName, YYYY-MM-DD, City, Country".
2. The script should filter the lines to only include people who are from a city that contains two or more words (e.g., "San Francisco"). Use a regular expression to check this.

3. For each of the filtered lines, the script should extract the person's age in years, based on the date of birth. You'll need to use a regular expression to extract the date of birth and then calculate the age.
4. After all ages are calculated, sort the people by their ages in descending order and print each person's name along with their age.

Note: You can assume that the current year is 2024 for age calculations.