Planning:

Input: Path of the directory

Output: 2 text files in the input directory which contains all the words without duplicates and how many times each word appears in the set of words respectively.

- Start the program.
- Ask for home directory path.
- Files in the home directory and sub directories are traversed.
- A new file of name “words.txt” is created.
- Create another file called “words_histogram.txt”.
- The text files are filtered out.
- The contents of the text file are read.
- Use a dictionary to create a histogram.
- While reading each file, write each word into the “words.txt” file, without any duplicate words.
- Use the word as key and set count as 1 if the word is not present as key, or else increment the count by 1.
- After reading all the files, close the file “words.txt”.
- Write each key(word) and its value(frequency) into “words_histogram.txt” file.
- After writing close the file.
- End the program.

Design:

1) Start the program.
2) Import the os module.
3) Read home directory path from user.
4) Use the walk method of os module to list the files present in home directory and sub directories as well.
5) Create an empty dictionary and empty list.
6) Create 2 new files, “words.txt” and “words_historgram.txt” in write mode.
7) Iterate through each file from os.walk(path) and append a new path combining the root directory and each file name using join method of os module, in the list.
8) Open each file in the list in read mode.

9) Use read () function and split it using blank space.
10) Write the output after splitting, one by one into the “words.txt” file.
11) If each word is present as a key in dictionary, increment its count by one. Else set each word as its key and set its count as 1.
12) Repeat every word in every file.
13) Write the dictionary into “words_histogram.txt” file.
14) Close “words.txt” and “words_histogram.txt” files.
15) Stop the program.

Coding:

```
"""This module provides functionality for processing text files in a
directory and writing the frequency of each word to a file.

The code uses the os module to walk through the specified directory and
its subdirectories to find all the text files. It then reads each text
file,
splits the contents into individual words, and writes each word to a file
called "words.txt".

The code also keeps track of the frequency of each word by using a
dictionary.
If a word already exists in the dictionary, its count is incremented. If
not, it
is added to the dictionary with a count of 1.

Finally, the code writes the frequency of each word to a file
called "words-histogram.txt" in the format of "word - frequency".

When the code is executed, it prompts the user to enter a home directory
to search for text files, and then processes the text files and writes the
output files accordingly.


Original Author: Pranesh Kumar

Created on: 12 Apr 2023

Last Edited: 14 Apr 2023

"""

# Importing the os module to work with os dependent functionality
import os


def get_text_files(directory: str) -> list:
    """This function gets all the text files present in the
    home directory and other subdirectories recursively.

    Args:
        directory (str): home directory which is got from the user

    Returns:
        list: list of text files with its absolute path
```

```
    """
    textfiles = []
    for root, folders, files in os.walk(directory):
        for file in files:
            if file.endswith(".txt"):
                textfiles.append(os.path.join(root, file))
    return textfiles


def get_words_and_frequency_write(textfiles: list, output_file: str =
"words.txt") -> dict:
    """Gets the words and its corresponding frequency from the files passed
to the function.
    Also writes the words to the output_file, which is also passed.

    Args:
        textfiles (list): list of text files with its absolute path
        output_file (str, optional): name of the output file along with its
path. Defaults to "words.txt".

    Returns:
        dict: frequency of the words present in the files
    """
    with open(output_file, "w") as words_file:
        frequency: dict = {}
        for file in textfiles:
            with open(file, "r") as f:
                try:
                    filecontents = f.read()
                    words = filecontents.split()
                    for word in words:
                        if word.isalpha():
                            words_file.write(word + "\n")
                            if word in frequency.keys():
                                frequency[word] += 1
                            else:
                                frequency[word] = 1
                        else:
                            continue
                except UnicodeDecodeError:
                    pass
        return frequency


def write_word_histogram(frequency_of_words: dict, output_file: str =
"words-histogram.txt") -> None:
    """Writes the frequency of each word to the given output file.

    Args:
        frequency_of_words (dict): frequency of the words present in the
files
        output_file (str, optional): name of the output file along with its
path. Defaults to "words-histogram.txt".
    """
    with open(output_file, "w") as histogram_file:
        for word, count in frequency_of_words.items():
            histogram_file.write(word + " - " + str(count) + "\n")
    histogram_file.close()


# driver code
```

```
if __name__ == "__main__":
    # Prompt the user to enter a home directory to search for text files
    home_path = input("Enter the home directory to search: ")

    # Find all the text files in the specified directory and its
subdirectories
    text_files = get_text_files(home_path)

    # Gets the list of words and its corresponding frequency and writes the
words to a file
    frequencyOfWords = get_words_and_frequency_write(text_files)

    # Write the frequency of each word to a file
    write_word_histogram(frequencyOfWords)
```

Q2)

```
def gethistogram(file="words.txt"):
    freq = {}
    with open(file, "r") as filehandle:
        lines = filehandle.readlines()
        for line in lines:
            if line in freq.keys():
                freq[line] += 1
            else:
                freq[line] = 1
    filehandle.close()

    return freq


def findprefix(pref):
    freq = gethistogram()
    res = {}
    for name, count in freq.items():
        if name.startswith(pref):
            res[name] = count
    result = dict(sorted(res.items(), key=lambda x: x[1], reverse=True))

    for name in result.keys():
        print(name)


if __name__ == "__main__":
    while True:
        try:
            prefix = input("Enter prefix: ")
            findprefix(prefix)
        except EOFError:
            break
```