Problem Solving & Coding - Level I [Language Foundations] (File I/O)

Solve the following problems using computer with help of Python/C++/Java/C# language as means of communication.

Problem 1: Missing Comments

When one writes a function, it is generally a good idea to include a comment that outlines the function's purpose, its parameters and its return value. However, sometimes comments are forgotten, or left out by well-intentioned programmers that plan to write them later but then never get around to it. Create a python function *findMissingComments* that takes python source file as input and identifies functions that are not immediately preceded by a comment. For the purposes of this exercise, assume that any line that begins with def, followed by a space, is the beginning of a function definition. Assume that the comment character, #, will be the first character on the previous line when the function has a comment. Display the names of all of the functions that are missing comments, along with line number where the function definition is located.

Write a main program to demonstrate your function.

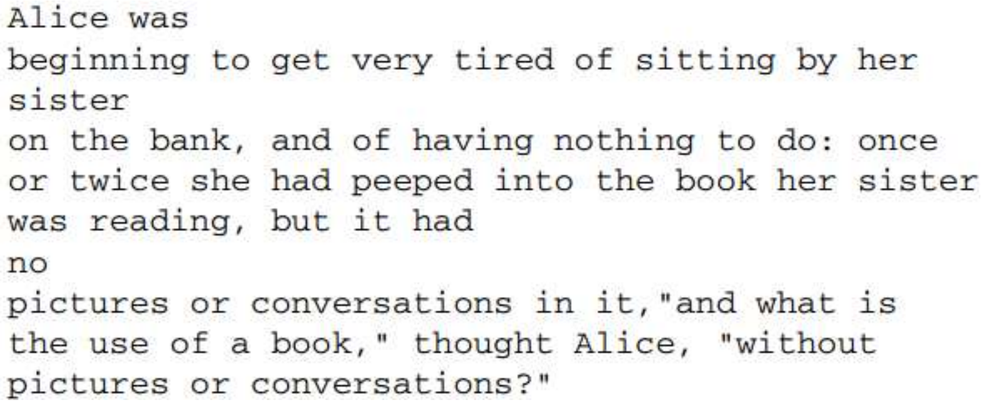
Problem 2: Consistent Length Formatting

While 80 characters is a common width for a terminal window, some terminals are narrow or wider. This can present challenges when displaying documents containing paragraphs of text. The lines might be too long and wrap, making them difficult to read, or they might be too short and fail to make use of the available space.

Create a function *formatText* that takes width, two filenames as input(one for input, one for output) and formats the text of first file to fit into given width ,and writes the formatted text to output file. If a line is too long then your program should break it up into words and add words to the current line until it is full. Then your program should start a new line and write the remaining words. Similarly, if a line is too short then you will need to use words from the next line of the file to finish filling the current line of output.

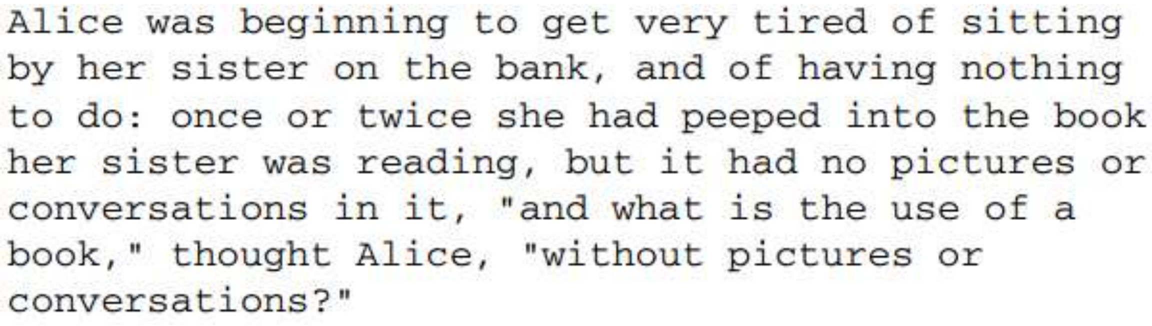
Write a main program to demonstrate your function. Ensure that your program works correctly for files containing multiple paragraphs of text. You can detect the end of one paragraph and the beginning of the next by looking for lines that are empty once the end of line marker has been removed.

For example, consider a file containing the following lines from “Alice’s Adventures in Wonderland”:



```
Alice was
beginning to get very tired of sitting by her
sister
on the bank, and of having nothing to do: once
or twice she had peeped into the book her sister
was reading, but it had
no
pictures or conversations in it,"and what is
the use of a book," thought Alice, "without
pictures or conversations?"
```

When formatted for a line length of 50 characters, it should be displayed as:



```
Alice was beginning to get very tired of sitting
by her sister on the bank, and of having nothing
to do: once or twice she had peeped into the book
her sister was reading, but it had no pictures or
conversations in it, "and what is the use of a
book," thought Alice, "without pictures or
conversations?"
```