LONGEST CHAIN PYRAMID

- You have a list of words, and you want to construct a pyramidal structure from them following these rules:

- Each row in the pyramid must consist of a word that contains one of the words from the row directly above it. (In other words, a word in a lower row must include all the letters of a word from the row above it.)

-Each word in a lower row can only have one additional letter compared to the word it extends from in the row above.

-The first row of the pyramid must consist of exactly one word, and that word should be the shortest in the list.

-The goal is to determine the maximum number of rows in such a pyramid.

-Recursive Function (build_pyramid): This function attempts to build a pyramid starting from the given current_word. It explores all other words that are one letter longer and checks if they contain the current_word as a subsequence.(You must use recursive approach).

Requirements:

-Write a function called longest_pyramid. This function should take a list of words and find the longest pyramid that can be constructed according to the above rules.

-Additionally, write a function called print_pyramid. This function should take the pyramid structure and print it in a formatted way, ensuring that the pyramid is properly aligned.

-Word Relationship: A word can "extend" another word if it contains the same letters in order. For example, the word "a" is contained within "ab", and "ab" is contained within "abc".

-Word Length: Each word in a lower row must be exactly one character longer than the word it extends from in the row above.

Example:

Input:words = ["a", "ab", "abc", "abcd", "abcde"]

Output:Maximum Pyramid Height: 5

Pyramid:

a

ab

abc

abcd

abcde

Example-2:

Input: words= ["x", "xy", "xyz", "xyztq"]

Output:Maximum Pyramid Height: 3

Pyramid:

x

xy

xyz

Example-3:

Input: words = ["a", "at", "cat", "scat", "scatter"]

Output: Maximum Pyramid Height:3

Pyramid:

a

at

cat