

1. Given a set of strings, find their longest common prefix (LCP).

Input: ["technique", "technician", "technology", "technical"]

Output: "techn"

Input: ["techie delight", "tech", "techie", "technology", "technical"]

Output: "tech"

2. Given a string S, find the longest palindromic substring in S. Substring of string S: $S[i \dots j]$ where $0 \leq i \leq j < \text{len}(S)$. Palindrome string: A string which reads the same backwards. More formally, S is palindrome if $\text{reverse}(S) = S$. In case of conflict, return the substring which occurs first (with the least starting index).

Example 1:

Input:

S = "aaaabbaa"

Output: aabbaa

Explanation: The longest Palindromic substring is "aabbaa".

Example 2:

Input:

S = "abc"

Output: a

Explanation: "a", "b" and "c" are the longest palindromes with same length.

The result is the one with the least starting index.

Expected Time Complexity: $O(|S|^2)$.

Expected Auxiliary Space: $O(1)$.

-
3. Find the anagram words (i.e. which have same set of characters) in the given sentence.

Input: How can you listen when you remain silent?

Output: listen, silent

-
4. Separate the given sentence with new line character whenever
- Space comes
 - Special character comes
 - Upper case within a word comes

Input: "This is a ExampleSentence"

Output:

"

This

is

an

Example
Sentence

"