Word Pattern

For this problem we must check if the given pattern and string s form a bijection (one-to-one mapping) between characters and words.

Key Idea:

- Split s into words by spaces.
- Check if each character in pattern consistently maps to the same word and vice versa.
- Ensure bijection:
  - Each character maps to exactly one word.
  - Each word maps to exactly one character.

Algorithm:

1. Split s by spaces → get vector of words.
2. If pattern.size() != words.size() → return false.
3. Use two hash maps:
   - charToWord: map pattern character → word
   - wordToChar: map word → pattern character
4. Iterate:
   - If mapping exists, check consistency.
   - If not, create new mapping.
5. If any mismatch → return false, else return true.

Complexity:

- Time: O(n) (n = number of words & pattern length)
- Space: O(n) for two maps.