Problem 1: Reverse Each Word in a String

Write a program that takes a string as input and reverses each word in the string while maintaining the original order of the words.

Example:

- **Input:** `"Hello World"`
- **Output:** `"olleH dlroW"`

Problem 2: Count Vowels in a String

Write a program that counts the number of vowels (a, e, i, o, u) in a given string, regardless of their case.

Example:

- **Input:** `"Programming is fun!"`
- **Output:** `6`

Problem 3: Remove Extra Spaces

Write a program that takes a string as input and removes any extra spaces between words, ensuring that only a single space exists between them. Also, trim any leading or trailing spaces.

Example:

- **Input:** `" Hello World "`
- **Output:** `"Hello World"`

Problem 4: Count Anagram Words in a String

Write a function that, given a string, counts and returns the number of words that are anagrams of a given word. A word in a string is separated by space(s).

Example:

- **Input:** `"listen silent enlist hello"`
- **Anagram Word:** `"listen"`
- **Output:** 3 (the words "listen," "silent," and "enlist" are anagrams of each other)

Problem 5: Find All Unique Words in a String

Write a function that takes a string as input and returns a list of all unique words (case insensitive) present in the string. Words are separated by space(s).

Example:

- **Input:** `"Hello world hello"`
- **Output:** `["hello", "world"]`

Problem 6: Rotate a String

Write a function that takes a string and an integer n, and rotates the string to the right by n positions.

Example:

- **Input:** "abcdef", 2
- **Output:** "efabcd"

Problem 7: Replace First and Last Character of Each Word in a String

Write a program that, given a string, replaces the first and last character of each word with a specified character.

Complete the function **replaceFirstLastChar()** that accepts a multi word string and a character, replacing the first and last character of every word in it with the specified character.

Note: Every two adjacent words in the given string will be separated by exactly one space character.

Example:

- **Input:** `"Hello World"`, `*`
- **Output:** `"*ell* *orl*"`