Python-2 Homework

1. **Swap First and Last Characters**
   Write a function to swap the first and last characters of a string. If length < 2, return the string as is.
   🌀 Input: `"python"` → Output: `"nythop"`

2. **Count Frequency of Each Vowel**
   Return a dictionary with counts of each vowel in the string.
   🔡 Input: `"education"` → Output: `{'a': 1, 'e': 1, 'i': 1, 'o': 1, 'u': 1}`

3. **Replace Digits with # in a String**
   Replace all digits in the string with the # character.
   🔢 Input: `"Room 101 is on floor 2"` → Output: `"Room ### is on floor #"`

4. **Compress the String (Run-Length Encoding)**
   Implement a basic RLE compression. Replace consecutive characters with `<char><count>` if count > 1.
   📦 Input: `"aaabbcddd"` → Output: `"a3b2cd3"`

5. **Check for Substring Rotation**
   Given two strings, check if one is a rotation of the other.
   🔄 Input: `"waterbottle", "erbottlewat"` → Output: `True`

6. **Find Most Frequent Word in a Sentence**
   Return the most frequently occurring word in a sentence. Ignore punctuation and case.
   📊 Input: `"The fox jumped over the lazy dog. The dog barked."` →

   Output: `"the"`

7. **Longest Palindromic Substring**
   Find the longest palindromic substring in a given string.
   🪞 Input: `"babad"` → Output: `"bab"` or `"aba"`

8. **Find All Words That Appear Exactly Once**
   Given a string of words separated by spaces, return a list of words that occur exactly once.
   🧠 Input: `"apple orange banana apple mango"` → Output: `["orange", "banana", "mango"]`

9. **Longest Substring Without Repeating Characters**
   Find the length of the longest substring without repeating characters.
   💡 Input: `"abcabcbb"` → Output: `3` (substring: `"abc"`)

10. **Generate All Valid Parentheses Strings of Length 2n**
    Given `n`, generate all valid combinations of `n` pairs of parentheses using recursion.
    ⚙️ Input: `n = 3` → Output: `["((()))", "(()())", "(())()", "()(())", "()()()"]`