```
"""
Fibonacci Series:
0 1 1 2 3 5 8 13 21 34...............

1. Print the first n fibonacci Series....
Ex:
Input: 5
Output: 0 1 1 2 3
Input: 10
Output: 0 1 1 2 3 5 8 13 21 34

def print_fibonacci_series(num):
    a = 0
    b = 1
    print(a, b, end=' ')
    for i in range(num - 2):
        c = a + b
        print(c, end=' ')
        a = b
        b = c


num = int(input("Enter N: "))
print_fibonacci_series(num)
"""


"""
Print the nth fibonacci number
Ex:
Input: 5
Output: 3
Input: 10
Output: 34

def calculate_nth_fibonacci_number(num):
    a = 0
    b = 1
    if num == 1:
        return a
    elif num == 2:
        return b
    for i in range(num - 2):
        c = a + b
        a = b
        b = c
    return c


num = int(input("Enter N: "))
print(calculate_nth_fibonacci_number(num))
"""


"""
Check if two strings are anagrams
Anagrams:
Two words are called anagrams if they contain the same characters same number of
times.
```

```
Ex:
bat, tab
cat, act
"""
def are_anagrams(word1: str, word2: str) -> bool:
    word1_chars = {}
    for char in word1:
        word1_chars[char] = word1_chars.get(char, 0) + 1

    word2_chars = {}
    for char in word2:
        word2_chars[char] = word2_chars.get(char, 0) + 1

    if len(word1_chars) != len(word2_chars):
        return False

    for char, char_count in word1_chars.items():
        if word2_chars.get(char, 0) != char_count:
            return False
    return True


word1 = input("Enter Word1: ")
word2 = input("Enter Word2: ")
if are_anagrams(word1, word2):
    print(f"{word1} and {word2} are anagrams...")
else:
    print(f"{word1} and {word2} are not anagrams...")
```