**Open-Ended Lab Task 2:**

**Objective:**

Given a *pattern* and a string *s*, find if *s* follows the same pattern.

Here **follow** means a full match, such that there is a bijection between a letter in *pattern* and a **non-empty** word in *s*. Specifically:

* Each letter in *pattern* maps to **exactly** one unique word in *s*.
* Each unique word in *s* maps to **exactly** one letter in *pattern*.
* No two letters map to the same word, and no two words map to the same letter.

**Submission Requirements:**

* Submit the program code along with comments explaining key sections of your implementation.
* Optional: Submit screenshots of sample outputs or visualizations if implemented.

**Examples:**

**Example 1:**

**Input:** pattern = "abba", s = "dog cat cat dog"

**Output:** true

**Explanation:**

The bijection can be established as:

* 'a' maps to "dog".
* 'b' maps to "cat".

**Example 2:**

**Input:** pattern = "abba", s = "dog cat cat fish"

**Output:** false

**Example 3:**

**Input:** pattern = "aaaa", s = "dog cat cat dog"

**Output:** false

**CODE:**

def word\_pattern(pat, s):

words = s.split()

if len(pat) != len(words):

return False

charac\_to\_word = {}

word\_to\_charac = {}

for char, word in zip(pat, words):

if char in charac\_to\_word:

if charac\_to\_word[char] != word:

return False

else:

charac\_to\_word[char] = word

if word in word\_to\_charac:

if word\_to\_charac[word] != char:

return False

else:

word\_to\_charac[word] = char

return True

if \_\_name\_\_ == "\_\_main\_\_":

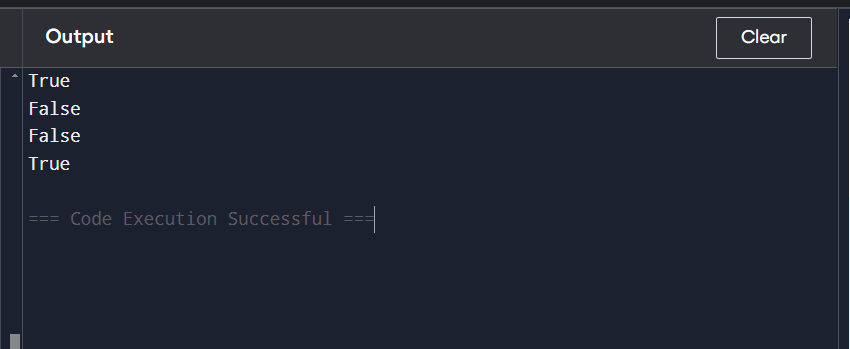
print(word\_pattern("abba", "dog cat cat dog")) # Output: True

print(word\_pattern("abba", "dog cat cat fish")) # Output: False

print(word\_pattern("aaaa", "dog cat cat dog")) # Output: False

print(word\_pattern("abc", "dog cat fish")) # Output: True

**OUTPUT:**

****