1. Python | Convert a list of Tuples into Dictionary

def convert\_tuples\_to\_dict(tuple\_list):

dictionary = dict(tuple\_list)

return dictionary

# Example usage

my\_tuples = [('a', 1), ('b', 2), ('c', 3)]

my\_dict = convert\_tuples\_to\_dict(my\_tuples)

print(my\_dict) # Output: {'a': 1, 'b': 2, 'c': 3}

1. Python counter and dictionary intersection example (Make a string using deletion and rearrangement)

from collections import Counter

def can\_make\_string(original\_string, target\_string):

original\_counts = Counter(original\_string)

target\_counts = Counter(target\_string)

intersection = original\_counts & target\_counts

return len(target\_string) == sum(intersection.values())

# Example usage

original = "abcdef"

target = "fedcba"

print(can\_make\_string(original, target)) # Output: True

1. Python dictionary, set and counter to check if frequencies can become same\

from collections import Counter

def can\_frequencies\_become\_same(string):

frequencies = Counter(string)

values = set(frequencies.values())

return len(values) == 1

# Example usage

my\_string = "aabbc"

print(can\_frequencies\_become\_same(my\_string)) # Output: True

1. Scraping And Finding Ordered Words In A Dictionary using Python

def find\_ordered\_words(dictionary, pattern):

ordered\_words = []

for word in dictionary:

if pattern in word:

ordered\_words.append(word)

return ordered\_words

# Example usage

my\_dictionary = ["apple", "banana", "cherry", "grape", "pineapple"]

my\_pattern = "ple"

ordered\_words = find\_ordered\_words(my\_dictionary, my\_pattern)

print(ordered\_words) # Output: ['apple', 'pineapple']

1. Possible Words using given characters in Python

from itertools import permutations

def find\_possible\_words(characters):

possible\_words = []

for length in range(1, len(characters) + 1):

permutations\_list = permutations(characters, length)

possible\_words.extend([''.join(word) for word in permutations\_list])

return possible\_words

# Example usage

my\_characters = "abc"

possible\_words = find\_possible\_words(my\_characters)

print(possible\_words) # Output: ['a', 'b', 'c', 'ab', 'ba', 'ac', 'ca', 'bc', 'cb', 'abc', 'acb', 'bac', 'bca', 'cab', 'cba']

1. Python – Keys associated with Values in Dictionary
2. Python program to Find the size of a Tuple
3. Python – Maximum and Minimum K elements in Tuple
4. Create a list of tuples from given list having number and its cube in each tuple
5. Python – Adding Tuple to List and vice – versa
6. Python – Closest Pair to Kth index element in Tuple