1. Write a Python program to find words which are greater than given length k?

def find\_words\_greater\_than\_k(word\_list, k):

result = []

for word in word\_list:

if len(word) > k:

result.append(word)

return result

# Example usage

words = ["apple", "banana", "cherry", "orange", "kiwi", "melon"]

k = 5

greater\_words = find\_words\_greater\_than\_k(words, k)

print("Words greater than length", k, ":", greater\_words)

1. Write a Python program for removing i-th character from a string?

def remove\_ith\_character(string, i):

if i < 0 or i >= len(string):

return string # Return the original string if the index is out of bounds

else:

return string[:i] + string[i+1:]

# Example usage

text = "Hello, World!"

index = 7

new\_text = remove\_ith\_character(text, index)

print("Original Text:", text)

print("After removing i-th character:", new\_text)

1. Write a Python program to split and join a string?

def split\_and\_join(string):

# Split the string into a list of words using whitespace as the delimiter

words = string.split()

# Join the words back into a string using a hyphen as the separator

joined\_string = "-".join(words)

return joined\_string

# Example usage

text = "Hello world! This is a sample string."

split\_text = split\_and\_join(text)

print("Original Text:", text)

print("After Splitting and Joining:", split\_text)

1. Write a Python to check if a given string is binary string or not?

def is\_binary\_string(string):

# Check if each character in the string is either '0' or '1'

for char in string:

if char != '0' and char != '1':

return False

return True

# Example usage

string1 = "101010"

string2 = "101012"

print(string1, "is binary:", is\_binary\_string(string1))

print(string2, "is binary:", is\_binary\_string(string2))

1. Write a Python program to find uncommon words from two Strings?

def find\_uncommon\_words(string1, string2):

# Split the strings into words

words1 = string1.split()

words2 = string2.split()

# Create sets of unique words from each string

set1 = set(words1)

set2 = set(words2)

# Find uncommon words by taking the symmetric difference of the two sets

uncommon\_words = set1.symmetric\_difference(set2)

return uncommon\_words

# Example usage

string1 = "Python is a programming language"

string2 = "Java is another programming language"

uncommon\_words = find\_uncommon\_words(string1, string2)

print("Uncommon words:", uncommon\_words)

1. Write a Python to find all duplicate characters in string?

def find\_duplicate\_characters(string):

# Create an empty list to store duplicate characters

duplicates = []

# Create a set to track characters we have already encountered

seen = set()

# Iterate over each character in the string

for char in string:

# Check if the character is already in the set

if char in seen:

# If it is, it's a duplicate, so add it to the duplicates list

duplicates.append(char)

else:

# If it's not in the set, add it to the set

seen.add(char)

return duplicates

# Example usage

string = "Hello, World!"

duplicate\_characters = find\_duplicate\_characters(string)

print("Duplicate characters:", duplicate\_characters)

1. Write a Python Program to check if a string contains any special character?

import re

def has\_special\_characters(string):

# Regular expression pattern to match special characters

pattern = r'[^a-zA-Z0-9\s]'

# Use the re module to search for special characters in the string

match = re.search(pattern, string)

# If a match is found, the string contains special characters

if match:

return True

else:

return False

# Example usage

string = "Hello, World!"

contains\_special\_characters = has\_special\_characters(string)

print("Contains special characters:", contains\_special\_characters)