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

A. def find\_words\_greater\_than\_length(sentence, k):

words = sentence.split()

result = []

for word in words:

if len(word) > k:

result.append(word)

return result

# Example usage

sentence = "This is a sample sentence to test the program."

k = 4

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

print(f"Words greater than length {k}: {words\_greater\_than\_k}")

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

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

# Check if i is a valid index

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

print("Invalid index")

return string

# Convert the string to a list of characters

characters = list(string)

# Remove the i-th character

characters.pop(i)

# Convert the list back to a string

result = "".join(characters)

return result

# Example usage

string = "Hello, World!"

i = 7

new\_string = remove\_ith\_character(string, i)

print(f"Original string: {string}")

print(f"String after removing character at index {i}: {new\_string}")

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

def split\_string(string, delimiter):

# Split the string into a list of substrings using the specified delimiter

substrings = string.split(delimiter)

return substrings

def join\_strings(strings, delimiter):

# Join the list of strings into a single string using the specified delimiter

result = delimiter.join(strings)

return result

# Example usage

string = "Hello, World!"

delimiter = ","

substrings = split\_string(string, delimiter)

print(f"String after splitting: {substrings}")

joined\_string = join\_strings(substrings, delimiter)

print(f"String after joining: {joined\_string}")

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

A. def is\_binary\_string(string):

# Iterate over each character in the string

for char in string:

# Check if the character is not '0' or '1'

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

return False

return True

# Example usage

string1 = "101010"

string2 = "101A010"

is\_binary1 = is\_binary\_string(string1)

print(f"{string1} is a binary string: {is\_binary1}")

is\_binary2 = is\_binary\_string(string2)

print(f"{string2} is a binary string: {is\_binary2}")

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

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

# Split the strings into individual words

words1 = string1.split()

words2 = string2.split()

# Initialize an empty set to store the uncommon words

uncommon\_words = set()

# Create a frequency dictionary for words in string1

frequency = {}

for word in words1:

frequency[word] = frequency.get(word, 0) + 1

# Add words from string2 that are not in frequency dictionary

for word in words2:

if word not in frequency:

uncommon\_words.add(word)

# Create a frequency dictionary for words in string2

frequency = {}

for word in words2:

frequency[word] = frequency.get(word, 0) + 1

# Add words from string1 that are not in frequency dictionary

for word in words1:

if word not in frequency:

uncommon\_words.add(word)

return uncommon\_words

# Example usage

string1 = "Hello world! This is string one."

string2 = "Hello there! This is string two."

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

print("Uncommon words:", uncommon)

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

A. def find\_duplicate\_characters(string):

# Create an empty set to store duplicate characters

duplicates = set()

# Create a set to store characters that have been seen

seen = set()

# Iterate over each character in the string

for char in string:

# Check if the character has been seen before

if char in seen:

# If so, it is a duplicate, add it to the duplicates set

duplicates.add(char)

else:

# If not, add it to the seen set

seen.add(char)

return duplicates

# Example usage

string = "Hello, World!"

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

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

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

A. import re

def contains\_special\_character(string):

# Regular expression pattern to match special characters

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

# Check if the string contains any special character

if re.search(pattern, string):

return True

else:

return False

# Example usage

string1 = "Hello, World!"

string2 = "Hello123"

contains\_special1 = contains\_special\_character(string1)

print(f"{string1} contains special character: {contains\_special1}")

contains\_special2 = contains\_special\_character(string2)

print(f"{string2} contains special character: {contains\_special2}")