**ADITYA AMIN**

**ASSIGN : 11**

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

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

words = sentence.split() # Split the sentence into words

result = []

for word in words:

if len(word) > k: # Check if word length is greater than k

result.append(word) # Append the word to the result list

return result

# Input sentence

sentence = "The quick brown fox jumped over the lazy dog"

# Input length threshold

k = 4

# Call the function and print the result

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

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

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

def remove\_character\_at\_index(string, i):

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

# Check if the given index is out of range

print("Error: Index out of range")

return string

else:

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

# Input string

string = "Hello, world!"

# Input index to remove

i = 7

# Call the function and print the updated string

updated\_string = remove\_character\_at\_index(string, i)

print("String with character at index {} removed: {}".format(i, updated\_string))

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

def split\_and\_join\_string(string, delimiter):

# Split the string by the delimiter

substrings = string.split(delimiter)

# Join the substrings using the same delimiter

joined\_string = delimiter.join(substrings)

return joined\_string

# Input string

string = "Hello, world! How are you?"

# Input delimiter for splitting and joining

delimiter = " " # Space

# Call the function to split and join the string

joined\_string = split\_and\_join\_string(string, delimiter)

# Print the original string, the delimiter used, and the joined string

print("Original string: ", string)

print("Delimiter used: ", delimiter)

print("Joined string: ", joined\_string)

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

def is\_binary\_string(string):

# Set of valid binary characters

binary\_characters = {'0', '1'}

# Check if all characters in the string are valid binary characters

for char in string:

if char not in binary\_characters:

return False

return True

# Input string

string = "101010"

# Call the function to check if the string is a binary string

if is\_binary\_string(string):

print("{} is a binary string.".format(string))

else:

print("{} is not a binary string.".format(string))

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

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

# Split the strings into words using space as delimiter

words1 = string1.split()

words2 = string2.split()

# Convert the words to sets for easy set operations

set1 = set(words1)

set2 = set(words2)

# Find uncommon words using set symmetric difference operation

uncommon\_words = list(set1 ^ set2)

return uncommon\_words

# Input strings

string1 = "Hello, how are you?"

string2 = "Hi, how is it going?"

# Call the function to find uncommon words

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

# Print the uncommon words

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

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

def find\_duplicate\_characters(string):

# Convert the string to lowercase to ignore case sensitivity

string = string.lower()

# Create a dictionary to store character frequency

char\_frequency = {}

# Iterate through each character in the string

for char in string:

# Ignore spaces

if char != " ":

# If character is not already in dictionary, add it with value 1

# If character is already in dictionary, increment its value by 1

if char not in char\_frequency:

char\_frequency[char] = 1

else:

char\_frequency[char] += 1

# Create a list to store duplicate characters

duplicate\_chars = []

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

def has\_special\_characters(string):

# Define a list of special characters

special\_chars = "!@#$%^&\*()\_+=-{}[]|\\;:'\"<>,.?/`~"

# Iterate through each character in the string

for char in string:

# If character is found in the list of special characters, return True

if char in special\_chars:

return True

# If no special character is found, return False

return False

# Input string

string = "Hello, world!"

# Call the function to check if string contains any special characters

result = has\_special\_characters(string)

# Print the result

if result:

print("String contains special characters.")

else:

print("String does not contain special characters.")