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

def string\_k(k, str):

# create the empty string

string = []

# split the string where space is comes

text = str.split(" ")

# iterate the loop till every substring

for x in text:

# if length of current sub string

# is greater than k then

if len(x) > k:

# append this sub string in

# string list

string.append(x)

# return string list

return string

# Driver Program

k = 3

str = "geek for geeks"

print(string\_k(k, str))

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

def remove(string, i):

# Characters before the i-th indexed

# is stored in a variable a

a = string[: i]

# Characters after the nth indexed

# is stored in a variable b

b = string[i + 1:]

# Returning string after removing

# nth indexed character.

return a + b

# Driver Code

if \_\_name\_\_ == '\_\_main\_\_':

string = "geeksFORgeeks"

# Remove nth index element

i = 5

# Print the new string

print(remove(string, i))

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

def split\_string(string):

# Split the string based on space delimiter

list\_string = string.split(' ')

return list\_string

def join\_string(list\_string):

# Join the string based on '-' delimiter

string = '-'.join(list\_string)

return string

# Driver Function

if \_\_name\_\_ == '\_\_main\_\_':

string = 'Geeks for Geeks'

# Splitting a string

list\_string = split\_string(string)

print(list\_string)

# Join list of strings into one

new\_string = join\_string(list\_string)

print(new\_string)

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

def check(string):

# set function convert string

# into set of characters .

p = set(string)

# declare set of '0', '1' .

s = {'0', '1'}

# check set p is same as set s

# or set p contains only '0'

# or set p contains only '1'

# or not, if any one condition

# is true then string is accepted

# otherwise not .

if s == p or p == {'0'} or p == {'1'}:

print("Yes")

else:

print("No")

# driver code

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

string = "101010000111"

# function calling

check(string)

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

def UncommonWords(A, B):

# count will contain all the word counts

count = {}

# insert words of string A to hash

for word in A.split():

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

# insert words of string B to hash

for word in B.split():

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

# return required list of words

return [word for word in count if count[word] == 1]

# Driver Code

A = "Geeks for Geeks"

B = "Learning from Geeks for Geeks"

# Print required answer

print(UncommonWords(A, B))

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

def duplicate\_characters(string):

# Create an empty dictionary

chars = {}

# Iterate through each character in the string

for char in string:

# If the character is not in the dictionary, add it

if char not in chars:

chars[char] = 1

else:

# If the character is already in the dictionary, increment the count

chars[char] += 1

# Create a list to store the duplicate characters

duplicates = []

# Iterate through the dictionary to find characters with count greater than 1

for char, count in chars.items():

if count > 1:

duplicates.append(char)

return duplicates

# Test cases

print(duplicate\_characters("geeksforgeeks"))

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

import re

# Function checks if the string

# contains any special character

def run(string):

# Make own character set and pass

# this as argument in compile method

regex = re.compile('[@\_!#$%^&\*()<>?/\|}{~:]')

# Pass the string in search

# method of regex object.

if(regex.search(string) == None):

print("String is accepted")

else:

print("String is not accepted.")

# Driver Code

if \_\_name\_\_ == '\_\_main\_\_' :

# Enter the string

string = "Geeks$For$Geeks"

# calling run function

run(string)