Question1

Create a function that takes three parameters where:

* x is the start of the range (inclusive).
* y is the end of the range (inclusive).
* n is the divisor to be checked against.

Return an ordered list with numbers in the range that are divisible by the third parameter n. Return an empty list if there are no numbers that are divisible by n.

**Examples**

list\_operation(1, 10, 3) ➞ [3, 6, 9]

list\_operation(7, 9, 2) ➞ [8]

list\_operation(15, 20, 7) ➞ []

# Create a function that takes three parameters where:

# • x is the start of the range (inclusive).

# • y is the end of the range (inclusive).

# • n is the divisor to be checked against.

# Define a function, function1

# which gives the list of numbers divisible by n in the range a to b

def function1(a,b,c):

l1 = []

for i in range(a,b+1):

if i%c==0:

l1.append(i)

return l1

# Call the function, function1 with given inputs and check results

l1 = function1(1, 10, 3) # it should give output [3, 6, 9]

print(l1)

l1 = function1(7, 9, 2) # it should give output [8]

print(l1)

l1 = function1(15, 20, 7) # it should give output []

print(l1)

Question2

Create a function that takes in two lists and returns True if the second list follows the first list by **one** element, and False otherwise. In other words, determine if the second list is the first list shifted to the right by 1.

**Examples**

simon\_says([1, 2], [5, 1]) ➞ True

simon\_says([1, 2], [5, 5]) ➞ False

simon\_says([1, 2, 3, 4, 5], [0, 1, 2, 3, 4]) ➞ True

simon\_says([1, 2, 3, 4, 5], [5, 5, 1, 2, 3]) ➞ False

**Notes**

* Both input lists will be of the same length, and will have a minimum length of 2.
* The values of the 0-indexed element in the second list and the n-1th indexed element in the first list do not matter.

# Create a function that takes in two lists and returns True if the second list follows the first list

# by one element, and False otherwise.

# Define a function, function1 which takes 2 parameters as list objects

# and returns true or false based on comparison

def function1(l1,l2):

c = True

for i in range(len(l1)-1):

if l1[i] != l2[i+1]:

c = False

return c

# Try with different inputs given and print the result

l3 , l4 = [1, 2, 3, 4, 5] , [0, 1, 2, 3, 4] # it should be true for these 2 lists

print(function1(l3,l4))

l3, l4 = [1, 2], [5, 5] # it should be false for these 2 lists

print(function1(l3,l4))

l3, l4 = [1, 2, 3, 4, 5], [0, 1, 2, 3, 4] # it should be true for these 2 lists

print(function1(l3,l4))

l3, l4 = [1, 2, 3, 4, 5], [5, 5, 1, 2, 3] # it should be false for these 2 lists

print(function1(l3,l4))

Question3

A group of friends have decided to start a secret society. The name will be the first letter of each of their names, sorted in alphabetical order.

Create a function that takes in a list of names and returns the name of the secret society.

### Examples

society\_name(["Adam", "Sarah", "Malcolm"]) ➞ "AMS"

society\_name(["Harry", "Newt", "Luna", "Cho"]) ➞ "CHLN"

society\_name(["Phoebe", "Chandler", "Rachel", "Ross", "Monica", "Joey"])

# Create a function that takes in a list of names and returns the name of the secret society.

# Define a function, function1

def function1(l1):

k =""

l1.sort()

for i in l1:

k = k + str(i)[0]

return k

# Check function, function1 with multiple given inputs

# and check the results

l1 = ["Adam", "Sarah", "Malcolm"] # It should return "AMS"

print(function1(l1))

l1 = ["Harry", "Newt", "Luna", "Cho"] # it should return "CHLN"

print(function1(l1))

l1 = ["Phoebe", "Chandler", "Rachel", "Ross", "Monica", "Joey"] # it should return "CJMPRR"

print(function1(l1))

Question4

An isogram is a word that has no duplicate letters. Create a function that takes a string and returns either True or False depending on whether or not it's an "isogram".

**Examples**

is\_isogram("Algorism") ➞ True

is\_isogram("PasSword") ➞ False

# Not case sensitive.

is\_isogram("Consecutive") ➞ False

**Notes**

* Ignore letter case (should not be case sensitive).
* All test cases contain valid one word strings.

# Create a function that takes a string and returns either True or False

# depending on whether or not it's an "isogram".

# Ignore letter case (should not be case sensitive).

# Define a function, function1

def function1(s1):

b = True

for i in range(0,len(s1)):

for j in range(0,len(s1)):

if s1[i].lower() == s1[j].lower() and i != j:

b = False

return b

# Call function, function1 with different inputs given and check the results

string1 = "Algorism"

print(function1(string1)) # It should return True

string1 = "PasSword"

print(function1(string1)) # It should return False

string1 = "Consecutive"

print(function1(string1)) # It should return False

Question5

Create a function that takes a string and returns True or False, depending on whether the characters are in order or not.

### Examples

is\_in\_order("abc") ➞ True

is\_in\_order("edabit") ➞ False

is\_in\_order("123") ➞ True

is\_in\_order("xyzz") ➞ True

### Notes

You don't have to handle empty strings.

# Create a function that takes a string and returns True or False,

# depending on whether the characters are in order or not.

# Define a function, function1, which takes string as input parameter

# and returns True or False based on the condition

def function1(s1):

char\_arr = list(s1)

char\_arr.sort()

return char\_arr == list(s1)

# Check with different inputs and verify the result

s2 = "abc"

print(function1(s2)) # it should return True

s2 = "edabit"

print(function1(s2)) # it should return False

s2 = "123"

print(function1(s2)) # it should return True

s2 = "xyzz"

print(function1(s2)) # it should return True