1. Given two strings, create a function that returns the total number

of unique characters from the combined string.

Examples

count\_unique('apple', 'play') ➞ 5

# 'appleplay' has 5 unique characters:

# 'a', 'e', 'l', 'p', 'y'

'sore', 'zebra' ➞ 7

'a', 'soup' ➞ 5

1. Given three lists of integers: lst1, lst2, lst3,

return the sum of integers which are common in all three lists.

Examples

sum\_common([1, 2, 3], [5, 3, 2], [7, 3, 2]) ➞ 5

// 2 & 3 are common in all 3 lists.

sum\_common([1, 2, 2, 3], [5, 3, 2, 2], [7, 3, 2, 2]) ➞ 7

// 2, 2 & 3 are common in all 3 lists.

sum\_common([1], [1], [2]) ➞ 0

1. Write a function that takes a list of numbers and returns a

list with two elements:

The first element should be the sum of all even numbers in the list.

The second element should be the sum of all odd numbers in the list.

Example

[1, 2, 3, 4, 5, 6] ➞ [12, 9]

# 2 + 4 + 6 = 12 and 1 + 3 + 5 = 9

[-1, -2, -3, -4, -5, -6] ➞ [-12, -9]

[0, 0] ➞ [0, 0]

Notes

Count 0 as an even number.

1. A number is said to be Harshad if it's exactly divisible

by the sum of its digits. Create a function that determines

whether a number is a Harshad or not.

Examples

is\_harshad(75) ➞ False

# 7 + 5 = 12

# 75 is not exactly divisible by 12

is\_harshad(171) ➞ True

# 1 + 7 + 1 = 9

# 9 exactly divides 171

is\_harshad(481) ➞ True

is\_harshad(89) ➞ False

is\_harshad(516) ➞ True

is\_harshad(200) ➞ True

1. Given an input string, reverse the string word by word.

Examples

"the sky is blue" ➞ "blue is sky the"

" hello world! " ➞ "world! hello"

"a good example" ➞ "example good a"

Notes

A word is defined as a sequence of non-space characters.

The input string may contain leading or trailing spaces.

However, your reversed string should not contain leading or

trailing spaces.

1. Create a function that builds a word from the scrambled letters

contained in the first list. Use the second list to establish

each position of the letters in the first list.

Return a string from the unscrambled letters (that made-up the word).

Examples

word\_builder(['g', 'e', 'o'], [1, 0, 2]) ➞ 'ego'

word\_builder(['e', 't', 's', 't'], [3, 0, 2, 1]) ➞ 'test'

word\_builder(['b', 'e', 't', 'i', 'd', 'a'], [1, 4, 5, 0, 3, 2]) ➞ 'edabit'

1. Return a new set of identical items from two sets

Given:

set1 = {10, 20, 30, 40, 50}

set2 = {30, 40, 50, 60, 70}

Expected output:

{40, 50, 30}

1. Write a Python program to return a new set with unique items

from both sets by removing duplicates.

Given:

set1 = {10, 20, 30, 40, 50}

set2 = {30, 40, 50, 60, 70}

Expected output:

{70, 40, 10, 50, 20, 60, 30}

1. Given two Python sets, write a Python program to update

the first set with items that exist only in the first set

and not in the second set.

Given:

set1 = {10, 20, 30}

set2 = {20, 40, 50}

Expected output:

set1 {10, 30}

1. Given an input string, reverse the string word by word

(reversed word also).

Examples

"the sky is blue" ➞ "eulb si yks eht"

Notes

A word is defined as a sequence of non-space characters.

The input string may contain leading or trailing spaces. However, your reversed string should not contain leading or trailing spaces.