Question 1:

Write a program that calculates and prints the value according to the given formula:

Q = Square root of [(2 * C * D)/H]

Following are the fixed values of C and H:

C is 50. H is 30.

D is the variable whose values should be input to your program in a comma-separated sequence.

Example

Let us assume the following comma separated input sequence is given to the program:

100,150,180

The output of the program should be:

18,22,24

Question 2:

Write a program which takes 2 digits, X,Y as input and generates a 2-dimensional array. The element value in the i-th row and j-th column of the array should be i*j.

Note: i=0,1.., X-1; j=0,1,¡Y-1.

Example

Suppose the following inputs are given to the program:

3,5

Then, the output of the program should be:

[[0, 0, 0, 0, 0], [0, 1, 2, 3, 4], [0, 2, 4, 6, 8]]

Question 3:

Write a program that accepts a comma separated sequence of words as input and prints the words in a comma-separated sequence after sorting them alphabetically.

Suppose the following input is supplied to the program:

without,hello,bag,world

Then, the output should be:

bag,hello,without,world

Question 4:

Write a program that accepts a sequence of whitespace separated words as input and prints the words after removing all duplicate words and sorting them alphanumerically.

Suppose the following input is supplied to the program:

hello world and practice makes perfect and hello world again

Then, the output should be:

again and hello makes perfect practice world

Question 5:

Write a program that accepts a sentence and calculate the number of letters and digits.

Suppose the following input is supplied to the program:

hello world! 123

Then, the output should be:

LETTERS 10

DIGITS 3

Question 6:

A website requires the users to input username and password to register. Write a program to check the validity of password input by users.

Following are the criteria for checking the password:

1. At least 1 letter between [a-z]

2. At least 1 number between [0-9]

1. At least 1 letter between [A-Z]

3. At least 1 character from [$#@]

4. Minimum length of transaction password: 6

5. Maximum length of transaction password: 12

Your program should accept a sequence of comma separated passwords and will check them according to the above criteria. Passwords that match the criteria are to be printed, each separated by a comma.

Example

If the following passwords are given as input to the program:

ABd1234@1,a F1#,2w3E*,2We3345

Then, the output of the program should be:

ABd1234@1

Question 7:

Define a class with a generator which can iterate the numbers, which are divisible by 7, between a given range 0 and n.

Question 8:

Write a program to compute the frequency of the words from the input. The output should output after sorting the key alphanumerically.

Suppose the following input is supplied to the program:

New to Python or choosing between Python 2 and Python 3? Read Python 2 or Python 9.

Then, the output should be:

2:2

3.:1

3?:1

New:1

Python:5

Read:1

and:1

between:1

choosing:1

or:2

to:1

Question 9:

Define a class Person and its two child classes: Male and Female. All classes have a method "getGender" which can print "Male" for Male class and "Female" for Female class.

Question 10:

Please write a program to generate all sentences where subject is in ["I", "You"] and verb is in ["Play", "Love"] and the object is in ["Hockey", "Football"].

Question 11:

Please write a program to compress and decompress the string "hello world!hello world!hello world!hello world!".

Question :12

Please write a binary search function which searches an item in a sorted list. The function should return the index of element to be searched in the list.

Question 13:

Please write a program using generator to print the numbers which can be divisible by 5 and 7 between 0 and n in comma separated form while n is input by console.

Example:

If the following n is given as input to the program:

100

Then, the output of the program should be:

0,35,70

Question 14:

Please write a program using generator to print the even numbers between 0 and n in comma separated form while n is input by console.

Example:

If the following n is given as input to the program:

10

Then, the output of the program should be:

0,2,4,6,8,10

Question 15:

The Fibonacci Sequence is computed based on the following formula:

f(n) = 0 if n = 0

f(n) = 1 if n = 1

f(n) = f(n-1) + f(n-2) if n > 1

Please write a program using list comprehension to print the Fibonacci Sequence in comma separated form with a given n input by console.

Example:

If the following n is given as input to the program:

7

Then, the output of the program should be:

0,1,1,2,3,5,8,13

Question 16:

Assuming that we have some email addresses in the "username@companyname.com" format, please write a program to print the user name of a given email address. Both user names and company names are composed of letters only.

Example:

If the following email address is given as input to the program:

john@google.com

Then, the output of the program should be:

john

Question 17:

Define a class named Shape and its subclass Square. The Square class has an init function which takes a length as argument. Both classes have a area function which can print the area of the shape where Shape's area is 0 by default.

Question 18.

Write a function that stutters a word as if someone is struggling to read it. The first two letters are repeated twice with an ellipsis ... and space after each, and then the word is pronounced with a question mark ?.

Examples:

stutter("incredible") ➞ "in... in... incredible?"

stutter("enthusiastic") ⟶ "en... en... enthusiastic?"

stutter("outstanding") ⟶ "ou... ou... outstanding?"Hint :- Assume all input is in lower case and at least two characters long.

Question 19.

Create a function that takes an angle in radians and returns the corresponding

angle in degrees rounded to one decimal place.

Examples

radians_to_degrees(1) ⟶ 57.3

radians_to_degrees(20) ⟶ 1145.9

radians_to_degrees(50) ⟶ 2864.8

Question 20.

In this challenge, establish if a given integer num is a Curzon number. If 1 plus

2 elevated to num is exactly divisible by 1 plus 2 multiplied by num, then num is a Curzon

number.

Given a non-negative integer num, implement a function that returns True if num is a Curzon

number, or False otherwise.

Examples

is_curzon(5) ⟶ True

2 ** 5 + 1 = 33

2 * 5 + 1 = 11

33 is a multiple of 11

is_curzon(10) ⟶ False

2 ** 10 + 1 = 1025

2 * 10 + 1 = 21

1025 is not a multiple of 21

is_curzon(14) ⟶ True

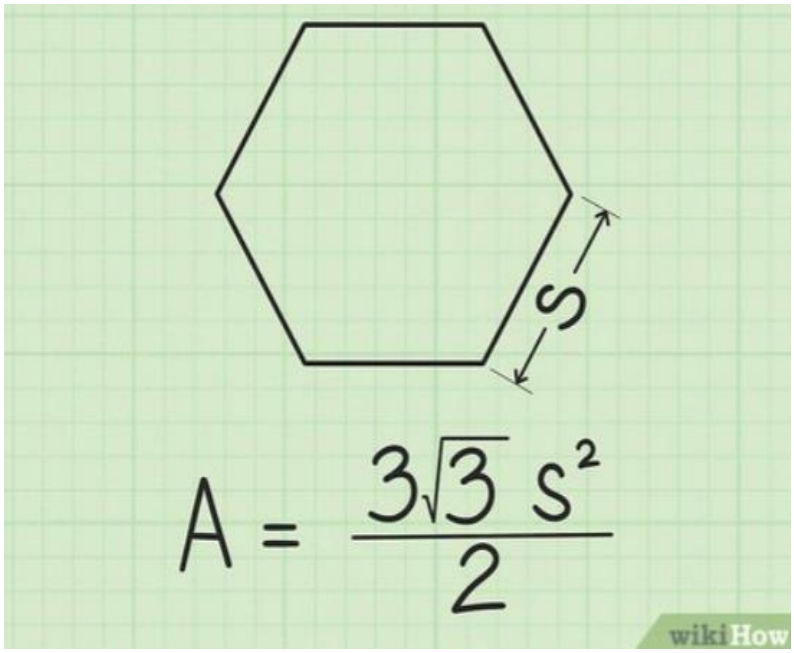
2 ** 14 + 1 = 16385

2 * 14 + 1 = 29

16385 is a multiple of 29

Question 21.

Given the side length x find the area of a hexagon.



Examples

area_of_hexagon(1) ➞ 2.6

area_of_hexagon(2) ➞ 10.4

area_of_hexagon(3) ➞ 23.4

Question 22.

Create a function that returns a base-2 (binary) representation of a base-10 (decimal) string number. To convert is simple: ((2) means base-2 and (10) means base-10)

010101001(2) = 1 + 8 + 32 + 128.

Going from right to left, the value of the most right bit is 1, now from that every bit to the left will be x2 the value, value of an 8 bit binary numbers are (256, 128, 64, 32, 16, 8, 4, 2, 1).

Examples:

binary(1) ➞ "1"

1*1 = 1

binary(5) ➞ "101"

11 + 14 = 5

binary(10) ➞ "1010"

12 + 18 = 10

Question 23.

Create a function that takes three arguments a, b, c and returns the sum of the numbers that are evenly divided by c from the range a, b inclusive.

Examples

evenly_divisible(1, 10, 20) ➞ 0

No number between 1 and 10 can be evenly divided by 20.

evenly_divisible(1, 10, 2) ➞ 30

2 + 4 + 6 + 8 + 10 = 30

evenly_divisible(1, 10, 3) ➞ 18

3 + 6 + 9 = 18

Question 24.

Create a function that returns True if a given inequality expression is correct and False otherwise.

Examples:

correct_signs("3 < 7 < 11") ➞ True

correct_signs("13 > 44 > 33 > 1") ➞ False

correct_signs("1 < 2 < 6 < 9 > 3") ➞ True

Question 25.

Create a function that replaces all the vowels in a string with a specified character.

Examples

replace_vowels("the aardvark", "#") ➞ "th# ##rdv#rk"

replace_vowels("minnie mouse", "?") ➞ "m?nn?? m??s?"

replace_vowels("shakespeare", "") ➞ "shksp**r"

Question 26.

Write a function that calculates the factorial of a number recursively.

Examples

factorial(5) ⟶ 120

factorial(3) ⟶ 6

factorial(1) ⟶ 1

factorial(0) ⟶ 1

Question 27.

Hamming distance is the number of characters that differ between two strings.

To illustrate:

String1: "abcbba"

String2: "abcbda"

Hamming Distance: 1 - "b" vs. "d" is the only difference.

Create a function that computes the hamming distance between two strings.

Examples:

hamming_distance("abcde", "bcdef") ⟶ 5

hamming_distance("abcde", "abcde") ⟶ 0

hamming_distance("strong", "strung") ⟶ 1

Question 28.

Create a function that takes a list of non-negative integers and strings and return a new list without the strings.

Examples

filter_list([1, 2, "a", "b"]) ⟶ [1, 2]

filter_list([1, "a", "b", 0, 15]) ⟶ [1, 0, 15]

filter_list([1, 2, "aasf", "1", "123", 123]) ⟶ [1, 2, 123]

Question 29.

The "Reverser" takes a string as input and returns that string in reverse order, with the opposite case.

Examples:

reverse("Hello World") ⟶ "DLROw OLLEh"

reverse("ReVeRsE") ⟶ "eSrEvEr"

reverse("Radar") ⟶ "RADAr"

Question 30.

You can assign variables from lists like this:

lst = [1, 2, 3, 4, 5, 6]

first = lst[0]

middle = lst[1:-1]

last = lst[-1]

print(first) ⟶ outputs 1

print(middle) ⟶ outputs [2, 3, 4, 5]

print(last) ⟶ outputs 6

With Python 3, you can assign variables from lists in a much more succinct way. Create variables first, middle and last from the given list using destructuring assignment (check the Resources tab for some examples), where:

first ⟶ 1

middle ⟶ [2, 3, 4, 5]

last ⟶ 6

Your task is to unpack the list writeyourcodehere into three variables, being first, middle, and last, with middle being everything in between the first and last element. Then print all three variables.

Question 31.

Write a function that calculates the factorial of a number recursively.

Examples

factorial(5) ⟶ 120

factorial(3) ⟶ 6

factorial(1) ⟶ 1

factorial(0) ⟶ 1

Question 32.

Write a function that moves all elements of one type to the end of the list.

Examples:

move_to_end([1, 3, 2, 4, 4, 1], 1) ⟶ [3, 2, 4, 4, 1, 1]

Move all the 1s to the end of the array.

move_to_end([7, 8, 9, 1, 2, 3, 4], 9) ⟶ [7, 8, 1, 2, 3, 4, 9]

move_to_end(["a", "a", "a", "b"], "a") ⟶ ["b", "a", "a", "a"]

Question 33.

Create a function that takes a string and returns a string in which each character is repeated once.

Examples:

double_char("String") ⟶ "SSttrriinngg"

double_char("Hello World!") ⟶ "HHeellllloo WWoorrlldd!!"

double_char("1234!_ ") ⟶ "11223344!!__ "

Question 34.

Create a function that reverses a boolean value and returns the string "boolean expected" if another variable type is given.

Examples:

reverse(True) ⟶ False

reverse(False) ⟶ True

reverse(0) ⟶ "boolean expected"

reverse(None) ⟶ "boolean expected"

Question 35.

Create a function that returns the thickness (in meters) of a piece of paper after folding it n number of times. The paper starts off with a thickness of 0.5mm.

Examples:

num_layers(1) ⟶ "0.001m"

Paper folded once is 1mm (equal to 0.001m)

num_layers(4) ⟶ "0.008m"

Paper folded 4 times is 8mm (equal to 0.008m)

num_layers(21) ⟶ "1048.576m"

Paper folded 21 times is 1048576mm (equal to 1048.576m)

Question 36.

Create a function that takes a single string as argument and returns an ordered list containing the indices of all capital letters in the string.

Examples

index_of_caps("eDaBiT") ⟶ [1, 3, 5]

index_of_caps("eQuINoX") ⟶ [1, 3, 4, 6]

index_of_caps("determine") ⟶ []

index_of_caps("STRIKE") ⟶ [0, 1, 2, 3, 4, 5]

index_of_caps("sUn") ⟶ [1]

Question 37.

Using list comprehensions, create a function that finds all even numbers from 1 to the given

number.

Examples

find_even_nums(8) ⟶ [2, 4, 6, 8]

find_even_nums(4) ⟶ [2, 4]

find_even_nums(2) ⟶ [2]

Question 38.

Create a function that takes a list of strings and integers, and filters out the list so that it

returns a list of integers only.

Examples

filter_list([1, 2, 3, "a", "b", 4]) ⟶ [1, 2, 3, 4]

filter_list(["A", 0, "Edabit", 1729, "Python", "1729"]) ⟶ [0, 1729]

filter_list(["Nothing", "here"]) ⟶ []

Question 39.

Given a list of numbers, create a function which returns the list but with each element's index in the list added to itself. This means you add 0 to the number at index 0, add 1 to the number at index 1, etc...

Examples:

add_indexes([0, 0, 0, 0, 0]) ⟶ [0, 1, 2, 3, 4]

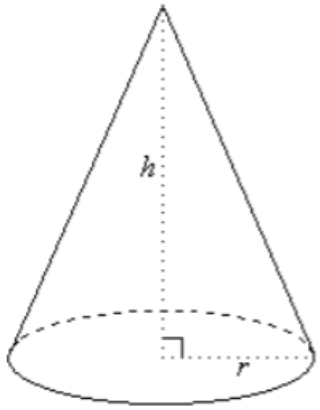
add_indexes([1, 2, 3, 4, 5]) ⟶ [1, 3, 5, 7, 9]

add_indexes([5, 4, 3, 2, 1]) ⟶ [5, 5, 5, 5, 5]

Question 40.

Create a function that takes the height and radius of a cone as arguments and returns the

volume of the cone rounded to the nearest hundredth. See the resources tab for the formula.



Examples

cone_volume(3, 2) ➞ 12.57

cone_volume(15, 6) ➞ 565.49

cone_volume(18, 0) ➞ 0

Question 41.

This Triangular Number Sequence is generated from a pattern of dots that form a triangle.

The first 5 numbers of the sequence, or dots, are:

1, 3, 6, 10, 15

This means that the first triangle has just one dot, the second one has three dots, the third one has 6 dots and so on.

Write a function that gives the number of dots with its corresponding triangle number of the sequence.

Examples

triangle(1) ➞ 1

triangle(6) ➞ 21

triangle(215) ➞ 23220

Question 42.

Create a function that takes a list of numbers between 1 and 10 (excluding one number) and returns the missing number.

Examples

missing_num([1, 2, 3, 4, 6, 7, 8, 9, 10]) ➞ 5

missing_num([7, 2, 3, 6, 5, 9, 1, 4, 8]) ➞ 10

missing_num([10, 5, 1, 2, 4, 6, 8, 3, 9]) ➞ 7

Question 43.

Write a function that takes a list and a number as arguments. Add the number to the end of the list, then remove the first element of the list. The function should then return the updated list.

Examples

next_in_line([5, 6, 7, 8, 9], 1) ➞ [6, 7, 8, 9, 1]

next_in_line([7, 6, 3, 23, 17], 10) ➞ [6, 3, 23, 17, 10]

next_in_line([1, 10, 20, 42], 6) ➞ [10, 20, 42, 6]

next_in_line([], 6) ➞ "No list has been selected"

Question 44.

Create the function that takes a list of dictionaries and returns the sum of people's budgets.

Examples

get_budgets([

{ "name": "John", "age": 21, "budget": 23000 },

{ "name": "Steve", "age": 32, "budget": 40000 },

{ "name": "Martin", "age": 16, "budget": 2700 }

]) ➞ 65700

get_budgets([

{ "name": "John", "age": 21, "budget": 29000 },

{ "name": "Steve", "age": 32, "budget": 32000 },

{ "name": "Martin", "age": 16, "budget": 1600 }

]) ➞ 62600

Question 46.

Create a function that takes a string and returns a string with its letters in alphabetical order.

Examples

alphabet_soup("hello") ➞ "ehllo"

alphabet_soup("edabit") ➞ "abdeit"

alphabet_soup("hacker") ➞ "acehkr"

alphabet_soup("geek") ➞ "eegk"

alphabet_soup("javascript") ➞ "aacijprstv"

Question 47.

Suppose that you invest $10,000 for 10 years at an interest rate of 6% compounded monthly. What will be the value of your investment at the end of the 10 year period?

Create a function that accepts the principal p, the term in years t, the interest rate r, and the number of compounding periods per year n. The function returns the value at the end of term rounded to the nearest cent.

For the example above:

compound_interest(10000, 10, 0.06, 12) ➞ 18193.97

Note that the interest rate is given as a decimal and n=12 because with monthly compounding there are 12 periods per year. Compounding can also be done annually, quarterly, weekly, or daily.

Examples

compound_interest(100, 1, 0.05, 1) ➞ 105.0

compound_interest(3500, 15, 0.1, 4) ➞ 15399.26

compound_interest(100000, 20, 0.15, 365) ➞ 2007316.26

Question 48.

Write a function that takes a list of elements and returns only the integers.

Examples

return_only_integer([9, 2, "space", "car", "lion", 16]) ➞ [9, 2, 16]

return_only_integer(["hello", 81, "basketball", 123, "fox"]) ➞ [81, 123]

return_only_integer([10, "121", 56, 20, "car", 3, "lion"]) ⟶ [10, 56, 20, 3]

return_only_integer(["String", True, 3.3, 1]) ⟶ [1]

Question 49.

Write a function that takes a list and a number as arguments. Add the number to the end of the list, then remove the first element of the list. The function should then return the updated list.

Examples

next_in_line([5, 6, 7, 8, 9], 1) ⟶ [6, 7, 8, 9, 1]

next_in_line([7, 6, 3, 23, 17], 10) ⟶ [6, 3, 23, 17, 10]

next_in_line([1, 10, 20, 42], 6) ⟶ [10, 20, 42, 6]

next_in_line([], 6) ⟶ "No list has been selected"

Question 50.

Create the function that takes a list of dictionaries and returns the sum of people's budgets.

Examples

get_budgets([

{ "name": "John", "age": 21, "budget": 23000 },

{ "name": "Steve", "age": 32, "budget": 40000 },

{ "name": "Martin", "age": 16, "budget": 2700 }

]) ⟶ 65700

get_budgets([

{ "name": "John", "age": 21, "budget": 29000 },

{ "name": "Steve", "age": 32, "budget": 32000 },

{ "name": "Martin", "age": 16, "budget": 1600 }

]) ⟶ 62600

Question 3:

Create a function that takes a string and returns a string with its letters in alphabetical order.

Examples

alphabet_soup("hello") ⟶ "ehllo"

alphabet_soup("edabit") ⟶ "abdeit"

alphabet_soup("hacker") ⟶ "acehkr"

alphabet_soup("geek") ⟶ "eegk"

alphabet_soup("javascript") ⟶ "aacijprstv"

Question 51.

Suppose that you invest $10,000 for 10 years at an interest rate of 6% compounded monthly. What will be the value of your investment at the end of the 10 year period?

Create a function that accepts the principal p, the term in years t, the interest rate r, and the number of compounding periods per year n. The function returns the value at the end of term rounded to the nearest cent.

For the example above:

compound_interest(10000, 10, 0.06, 12) ⟶ 18193.97

Note that the interest rate is given as a decimal and n=12 because with monthly compounding there are 12 periods per year. Compounding can also be done annually, quarterly, weekly, or daily.

Examples

compound_interest(100, 1, 0.05, 1) ⟶ 105.0

compound_interest(3500, 15, 0.1, 4) ⟶ 15399.26

compound_interest(100000, 20, 0.15, 365) ⟶ 2007316.26

Question 52.

Write a function that takes a list of elements and returns only the integers.

Examples

return_only_integer([9, 2, "space", "car", "lion", 16]) ⟶ [9, 2, 16]

return_only_integer(["hello", 81, "basketball", 123, "fox"]) ⟶ [81, 123]

return_only_integer([10, "121", 56, 20, "car", 3, "lion"]) ⟶ [10, 56, 20, 3]

return_only_integer(["String", True, 3.3, 1]) ⟶ [1]

Question 53.

Create a function that takes an integer and returns a list from 1 to the given number, where:

If the number can be divided evenly by 4, amplify it by 10 (i.e. return 10 times the number).

If the number cannot be divided evenly by 4, simply return the number.

Examples

amplify(4) ➞ [1, 2, 3, 40]

amplify(3) ➞ [1, 2, 3]

amplify(25) ➞ [1, 2, 3, 40, 5, 6, 7, 80, 9, 10, 11, 120, 13, 14, 15, 160, 17, 18, 19, 200, 21, 22, 23, 240, 25]

Notes

The given integer will always be equal to or greater than 1.

Include the number (see example above).

To perform this problem with its intended purpose, try doing it with list comprehensions. If that's too difficult, just solve the challenge any way you can.

Question 54.

Create a function that takes a list of numbers and return the number that's unique.

Examples

unique([3, 3, 3, 7, 3, 3]) ➞ 7

unique([0, 0, 0.77, 0, 0]) ➞ 0.77

unique([0, 1, 1, 1, 1, 1, 1, 1]) ➞ 0

Notes

Test cases will always have exactly one unique number while all others are the same.

Question 3:

Your task is to create a Circle constructor that creates a circle with a radius provided by an argument. The circles constructed must have two getters getArea() (PIr^2) and getPerimeter() (2PI*r) which give both respective areas and perimeter (circumference).

For help with this class, I have provided you with a Rectangle constructor which you can use as a base example.

Examples

circy = Circle(11)

circy.getArea()

Should return 380.132711084365

circy = Circle(4.44)

circy.getPerimeter()

Should return 27.897342763877365

Notes

Round results up to the nearest integer.

Question 55.

Create a function that takes a list of strings and return a list, sorted from shortest to longest.

Examples

sort_by_length(["Google", "Apple", "Microsoft"])

➞ ["Apple", "Google", "Microsoft"]

sort_by_length(["Leonardo", "Michelangelo", "Raphael", "Donatello"])

➞ ["Raphael", "Leonardo", "Donatello", "Michelangelo"]

sort_by_length(["Turing", "Einstein", "Jung"])

➞ ["Jung", "Turing", "Einstein"]

Notes

All test cases contain lists with strings of different lengths, so you won't have to deal with multiple strings of the same length.

Question 56.

Create a function that validates whether three given integers form a Pythagorean triplet. The sum of the squares of the two smallest integers must equal the square of the largest number to be validated.

Examples

is_triplet(3, 4, 5) ➞ True

$3^2 + 4^2 = 25$

$5^2 = 25$

is_triplet(13, 5, 12) ➞ True

$5^2 + 12^2 = 169$

$13^2 = 169$

is_triplet(1, 2, 3) ⟶ False

$1^2 + 2^2 = 5$

$3^2 = 9$

Notes

Numbers may not be given in a sorted order.

Question 57.

Create a function that takes three integer arguments (a, b, c) and returns the amount of integers which are of equal value.

Examples

equal(3, 4, 3) ⟶ 2

equal(1, 1, 1) ⟶ 3

equal(3, 4, 1) ⟶ 0

Notes

Your function must return 0, 2 or 3.

Question 58.

Write a function that converts a dictionary into a list of keys-values tuples.

Examples

dict_to_list({

"D": 1,

"B": 2,

"C": 3

}) ⟶ [("B", 2), ("C", 3), ("D", 1)]

dict_to_list({

"likes": 2,

"dislikes": 3,

"followers": 10

}) ⟶ [("dislikes", 3), ("followers", 10), ("likes", 2)]

Notes

Return the elements in the list in alphabetical order.

Question 59.

Write a function that creates a dictionary with each (key, value) pair being the (lower case, upper case) versions of a letter, respectively.

Examples

mapping(["p", "s"]) ➞ { "p": "P", "s": "S" }

mapping(["a", "b", "c"]) ➞ { "a": "A", "b": "B", "c": "C" }

mapping(["a", "v", "y", "z"]) ➞ { "a": "A", "v": "V", "y": "Y", "z": "Z" }

Notes

All of the letters in the input list will always be lowercase.

Question 60.

Write a function, that replaces all vowels in a string with a specified vowel.

Examples

vow_replace("apples and bananas", "u") ➞ "upplus und bununus"

vow_replace("cheese casserole", "o") ➞ "chooso cossorolo"

vow_replace("stuffed jalapeno poppers", "e") ➞ "steffed jelepene peppers"

Notes

All words will be lowercase. Y is not considered a vowel.

Question 61.

Create a function that takes a string as input and capitalizes a letter if its ASCII code is even and returns its lower case version if its ASCII code is odd.

Examples

ascii_capitalize("to be or not to be!") ➞ "To Be oR NoT To Be!"

ascii_capitalize("THE LITTLE MERMAID") ➞ "THe LiTTLe meRmaiD"

ascii_capitalize("Oh what a beautiful morning.") ➞ "oH wHaT a BeauTiFuL moRNiNg."