Assignment

Functions (Part 2)

Question # 1: 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 # 2: 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 # 3: 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 # 4: 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 # 5: 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 # 6: Write a function that moves same elements 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']