Task (3)

Problem solving

1) Given a number, n, return a function which adds n to the number passed to it.

EX:

Add (10)(20) ⟶ 30

Add (0)(20) ⟶ 20

Add (-30)(80) ⟶ 50

2) Create a function that takes in a number as a string n and returns the number without trailing and leading zeros.

Trailing Zeros are the zeros after a decimal point ***which don't affect the value*** (e.g. the last three zeros in 3.4000 and 3.04000).

Leading Zeros are the zeros before a whole number ***which don't affect the value*** (e.g. the first three zeros in 000234 and 000230).

EX:

removeLeadingTrailing("230.000") ⟶ "230"

removeLeadingTrailing("00402") ⟶ "402"

removeLeadingTrailing("03.1400") ⟶ "3.14"

removeLeadingTrailing("30") ⟶ "30"

3)Create a function that takes an array of numbers and returns the second largest number.

EX:

secondLargest([10, 40, 30, 20, 50]) ⟶ 40

secondLargest([25, 143, 89, 13, 105]) ⟶ 105

secondLargest([54, 23, 11, 17, 10]) ⟶ 23

4) A repdigit is a **positive number** composed out of the same digit. Create a function that takes an integer and returns whether it's a repdigit or not.

EX:

isRepdigit(66) ⟶ true

isRepdigit(0) ⟶ true

isRepdigit(-11) ⟶ false

5) Given an input string, reverse the string word by word, the first word will be the last, and so on.

EX:

reverseWords(" the sky is blue") ⟶ "blue is sky the"

reverseWords("hello world! ") ⟶ "world! hello"

reverseWords("a good example") ⟶ "example good a"

6) Create a function that takes an array of numbers and return "Boom!" if the digit 7 appears in the array.

Otherwise, return "there is no 7 in the array".

EX:

sevenBoom([1, 2, 3, 4, 5, 6, 7]) ⟶ "Boom!"

➔ 7 contains the number seven.

sevenBoom([8, 6, 33, 100]) ⟶ "there is no 7 in the array"

➔ None of the items contain 7 within them.

sevenBoom([2, 55, 60, 97, 86]) ⟶ "Boom!"

➔ 97 contains the number seven.

7) Write a function that inserts a white space between every instance of a lower character followed immediately by an upper character.

EX:

insertWhitespace("SheWalksToTheBeach") ⟶ "She Walks To The Beach"

insertWhitespace("MarvinTalksTooMuch") ⟶ "Marvin Talks Too Much"

insertWhitespace("TheGreatestUpsetInHistory") ⟶ "The Greatest Upset In History"

8) Create a function which returns the number of true values there are in an array.

EX:

countTrue([true, false, false, true, false]) ⟶ 2

countTrue([false, false, false, false]) ⟶ 0

countTrue([]) ⟶ 0

9) Create a function that moves all capital letters to the front of a word.

EX:

capToFront("hApPy") ⟶ "APhpy"

capToFront("moveMENT") ⟶ "MENTmove"

capToFront("shOrtCAKE") ⟶ "OCAKEshrt"

10) Create a function that takes an array of items and checks if the last item matches the rest of the array concatenated together.

EX:

matchLastItem(["rsq", "6hi", "g", "rsq6hig"]) ⟶ true
→ The last item is the rest joined.

matchLastItem([1, 1, 1, "11"]) ⟶ false
→ The last item should be "111".

matchLastItem([8, "thunder", true, "8thundertrue"]) ⟶ true

11) Create a function to find NaN in an array of numbers. The return value should be the **index where NaN is found**. If NaN is not found in the array***, then return -1***.

EX:

findNaN([1, 2, NaN]) ➞ 2

findNaN([NaN, 1, 2, 3, 4]) ➞ 0

findNaN([0, 1, 2, 3, 4]) ➞ -1

12) Create a function that takes an array of items, removes all duplicate items and returns a new array in the same sequential order as the old array (minus duplicates).

EX:

removeDups([1, 0, 1, 0]) ➞ [1, 0]

removeDups(["The", "big", "cat"]) ➞ ["The", "big", "cat"]

removeDups(["John", "Taylor", "John"]) ➞ ["John", "Taylor"]

13) Write a function that receives the time in 12-hour AM/PM format and returns a string representation of the time in military (24-hour) format.

EX:

convertTime(“07:05:45PM”) ➞ “19:05:45”

convertTime(“12:40:22AM”) ➞ “00:40:22”

convertTime(“12:45:54PM”) ➞ “12:45:54”

14) Write a function that removes the last vowel in each word in a sentence.

Ex:

removeLastVowel("Those who dare to fail miserably can achieve greatly.")
➞ "Thos wh dar t fal miserbly cn achiev gretly."

removeLastVowel("Love is a serious mental disease.")
➞ "Lov s serios mentl diseas"

15) Create a function that takes in an array of numbers and returns the sum of its cubes.

EX:

sumOfCubes([1, 5, 9]) ➞ 855
➔ Since 1^3 + 5^3 + 9^3 = 1 + 125 + 729 = 855

sumOfCubes([3, 4, 5]) ➞ 216

sumOfCubes([2]) ➞ 8

sumOfCubes([]) ➞ 0