**Name:**

**Advanced Programming in Java**

**Lab Exercise 1.20.2022**

1. Create a function that takes a string as an argument and returns a coded (h4ck3r 5p34k) version of the string.

Examples

hacker\_speak("javascript is cool") ➞ "j4v45cr1pt 15 c00l"

hacker\_speak("programming is fun") ➞ "pr0gr4mm1ng 15 fun"

hacker\_speak("become a coder") ➞ "b3c0m3 4 c0d3r"

Notes

In order to work properly, the function should replace all 'a's with 4, 'e's with 3, 'i's with 1, 'o's with 0, and 's's with 5

1. A number is narcissistic when the sum of its digits, with each digit raised to the power of digits quantity, is equal to the number itself.

153 ➞ 3 digits ➞ 1³ + 5³ + 3³ = 1 + 125 + 27 = 153 ➞ Narcissistic

84 ➞ 2 digits ➞ 8² + 4² = 64 + 16 = 80 ➞ Not narcissistic

Given a positive integer n, implement a function that returns True if the number is narcissistic, and False if it's not.

Examples

is\_narcissistic(8208) ➞ True # 8⁴ + 2⁴ + 0⁴ + 8⁴ = 8208

is\_narcissistic(22) ➞ False #2² + 2² = 8

is\_narcissistic(9) ➞ True # 9¹ = 9

Notes

Trivially, any number in the 1-9 range is narcissistic and any two-digit number is not.

Curious fact: Only 88 numbers are narcissistic.

1. Create a function that takes a number a and finds the missing exponent x so that a when raised to the power of x is equal to b.

Examples

solve\_for\_exp(4, 1024) ➞ 5

solve\_for\_exp(2, 1024) ➞ 10

solve\_for\_exp(9, 3486784401) ➞ 10

Notes

a is raised to the power of what in order to equal b?

1. Create a function that takes a single word string and does the following:

* Concatenates inator to the end if the word ends with a consonant otherwise,
* concatenate -inator instead.

Adds the word length of the original word to the end, supplied with '000'.

The examples should make this clear.

Examples

inatorInator('Shrink') ➞ 'Shrinkinator 6000'

inatorInator('Doom') ➞ 'Doominator 4000'

inatorInator('EvilClone') ➞ 'EvilClone-inator 9000'

1. A pair of strings form a strange pair if both of the following are true:

The 1st string's first letter = 2nd string's last letter.

The 1st string's last letter = 2nd string's first letter.

Create a function that returns True if a pair of strings constitutes a strange pair, and False otherwise.

Examples

is\_strange\_pair("ratio", "orator") ➞ True

# "ratio" ends with "o" and "orator" starts with "o".

# "ratio" starts with "r" and "orator" ends with "r".

is\_strange\_pair("sparkling", "groups") ➞ True

is\_strange\_pair("bush", "hubris") ➞ False

is\_strange\_pair("", "") ➞ True

Notes

It should work on a pair of empty strings (they trivially share nothing).