**1. 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.**

def stutter(word:str):

    print(f"{word[:2]}... {word[:2]}... {word}?")

stutter('incredible')

stutter('enthusiastic')

stutter('outstanding')

in... in... incredible?

en... en... enthusiastic?

ou... ou... outstanding?

**2. Create a function that takes an angle in radians and returns the corresponding angle in degrees rounded to one decimal place.**

def radians\_to\_degrees(rads):

    return round(57.295779513 \* rads, 1)

print(radians\_to\_degrees(1))

print(radians\_to\_degrees(20))

print(radians\_to\_degrees(50))

57.3

1145.9

2864.8

**3. 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.**

def is\_curzon(n):

    if (1 + 2 \*\* n) % (1 + 2\*n) == 0:

        return True

    return False

print(is\_curzon(5)) # True

print(is\_curzon(10)) # False

print(is\_curzon(14)) # True

True

False

True

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

import math

def area\_of\_hexagon(length):

    area =  3 \* math.sqrt(3) \* 0.5 \* length \* length

    return round(area, 1)

print(area\_of\_hexagon(1))

print(area\_of\_hexagon(2))

print(area\_of\_hexagon(3))

2.6

10.4

23.4

**5. 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.**

def binary(n):

    return "{0:b}".format(int(n))

print(binary(1))

print(binary(5))

print(binary(10))

1

101

1010