**Question1. 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(&quot;incredible&quot;) ➞ &quot;in... in... incredible?&quot;**

**stutter(&quot;enthusiastic&quot;) ➞ &quot;en... en... enthusiastic?&quot;**

**stutter(&quot;outstanding&quot;) ➞ &quot;ou... ou... outstanding?&quot;**

**Hint :- Assume all input is in lower case and at least two characters long.**

def stutter(word):

return word[:2] + "... " + word[:2] + "... " + word + "?"

# here's an example of how to use the stutter function:

word1 = "incredible"

word2 = "enthusiastic"

word3 = "outstanding"

stuttered\_word1 = stutter(word1)

stuttered\_word2 = stutter(word2)

stuttered\_word3 = stutter(word3)

print(stuttered\_word1) # "in... in... incredible?"

print(stuttered\_word2) # "en... en... enthusiastic?"

print(stuttered\_word3) # "ou... ou... outstanding?"

**Question 2.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**

def radians\_to\_degrees(radians):

degrees = radians \* 180 / 3.14159265358979323846

return round(degrees, 1)

# here's an example of how to use the radians\_to\_degrees function:

rad1 = 1

rad2 = 20

rad3 = 50

deg1 = radians\_to\_degrees(rad1)

deg2 = radians\_to\_degrees(rad2)

deg3 = radians\_to\_degrees(rad3)

print(deg1) # 57.3

print(deg2) # 1145.9

print(deg3) # 2864.8

**Question 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.**

**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**

def is\_curzon(num):

return (2\*\*num + 1) % (2 \* num + 1) == 0

# here's an example of how to use the is\_curzon function:

num1 = 5

num2 = 10

num3 = 14

result1 = is\_curzon(num1)

result2 = is\_curzon(num2)

result3 = is\_curzon(num3)

print(result1) # True

print(result2) # False

print(result3) # True

**Question 4.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**

import math

def area\_of\_hexagon(side\_length):

area = (3 \* math.sqrt(3) \* (side\_length \*\* 2)) / 2

return round(area, 1)

# here's an example of how to use the area\_of\_hexagon function:

side\_length1 = 1

side\_length2 = 2

side\_length3 = 3

result1 = area\_of\_hexagon(side\_length1)

result2 = area\_of\_hexagon(side\_length2)

result3 = area\_of\_hexagon(side\_length3)

print(result1) # 2.6

print(result2) # 10.4

print(result3) # 23.4

**Question 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.**

**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) ➞ &quot;1&quot;**

**# 1\*1 = 1**

**binary(5) ➞ &quot;101&quot;**

**# 1\*1 + 1\*4 = 5**

**binary(10) ➞ &quot;1010&quot;**

**# 1\*2 + 1\*8 = 10**

def binary(decimal):

binary = ''

value = 256

while value > 0:

if decimal >= value:

binary += '1'

decimal -= value

else:

binary += '0'

value //= 2

return binary

print(binary(1)) # Output: '1'

print(binary(5)) # Output: '101'

print(binary(10)) # Output: '1010'