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("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.

ANS: def stutter(word):

first\_two\_letters = word[:2]

stuttered\_word = f"{first\_two\_letters}... {first\_two\_letters}... {word}?"

return stuttered\_word

# Test the function

if \_\_name\_\_ == "\_\_main\_\_":

print(stutter("incredible")) # Output: "in... in... incredible?"

print(stutter("enthusiastic")) # Output: "en... en... enthusiastic?"

print(stutter("outstanding")) # Output: "ou... ou... outstanding?"

Example usage:

stutter("incredible") ➞ "in... in... incredible?"

stutter("enthusiastic") ➞ "en... en... enthusiastic?"

stutter("outstanding") ➞ "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

ANS: import math

def radians\_to\_degrees(radians):

degrees = radians \* (180 / math.pi)

return round(degrees, 1)

# Test the function

if \_\_name\_\_ == "\_\_main\_\_":

print(radians\_to\_degrees(1)) # Output: 57.3

print(radians\_to\_degrees(20)) # Output: 1145.9

print(radians\_to\_degrees(50)) # Output: 2864.8

Example Usage:

radians\_to\_degrees(1) ➞ 57.3

radians\_to\_degrees(20) ➞ 1145.9

radians\_to\_degrees(50) ➞ 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

ANS: def is\_curzon(num):

numerator = 2\*\*num + 1

denominator = 2\*num + 1

return numerator % denominator == 0

# Test the function

if \_\_name\_\_ == "\_\_main\_\_":

print(is\_curzon(5)) # Output: True

print(is\_curzon(10)) # Output: False

print(is\_curzon(14)) # Output: True

Example usage:

is\_curzon(5) ➞ True

is\_curzon(10) ➞ False

is\_curzon(14) ➞ 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

ANS: To find the area of a regular hexagon, you can use the formula:

Area = (3 \* √3 \* x^2) / 2

where x is the side length of the hexagon.

Here's a Python function to calculate the area of a hexagon:

import math

def area\_of\_hexagon(x):

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

return round(area, 1)

# Test the function

if \_\_name\_\_ == "\_\_main\_\_":

print(area\_of\_hexagon(1)) # Output: 2.6

print(area\_of\_hexagon(2)) # Output: 10.4

print(area\_of\_hexagon(3)) # Output: 23.4

In this function, we calculate the area of the hexagon using the given formula and the `math.sqrt()` function to calculate the square root.

Example usage:

area\_of\_hexagon(1) ➞ 2.6

area\_of\_hexagon(2) ➞ 10.4

area\_of\_hexagon(3) ➞ 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) ➞ "1"

# 1\*1 = 1

binary(5) ➞ "101"

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

binary(10) ➞ "1010"

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

ANS: def binary(decimal\_number):

decimal\_number = int(decimal\_number)

binary\_string = ""

while decimal\_number > 0:

remainder = decimal\_number % 2

binary\_string = str(remainder) + binary\_string

decimal\_number //= 2

if not binary\_string:

binary\_string = "0"

return binary\_string

# Test the function

if \_\_name\_\_ == "\_\_main\_\_":

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

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

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

Example usage:

binary(1) ➞ "1"

binary(5) ➞ "101"

binary(10) ➞ "1010"