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: ##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 ?.  
#inputs  
word=input("enter a word")  
#function  
def shutter(word):  
 str=word[0]+word[1]  
 print(str,"....",str,".....",word,"?")  
#call   
shutter(word)

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: ##a function that takes an angle in radians and returns the corresponding angle in degrees  
## rounded to one decimal place.  
  
#input  
rad=float(input("enter a radian value"))  
#function  
def radtodeg(rad):  
 return(rad\*57.3)  
#call  
deg=radtodeg(rad)  
print("the degree value of radian value {} is " .format(rad),round(deg,1))

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

; ##establish if a given integer num is a Curzon number  
  
#input  
num=int(input("enter an integer value"))  
def curzon(num):  
 x=2\*\*num+1  
 y=2\*num+1  
 if (x%y==0):  
 return ("true")  
 else:  
 return ("false")  
def check(num):  
 if num>0:  
 print(curzon(num))  
 else:  
 print("run again and enter a non zero integer value ")  
check(num)

Ans

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: ##the side length x find the area of a hexagon.  
#inputs  
side\_hex=float(input("enter the side of hexagon"))  
def area\_hex(s):  
 import math  
 return(((3\*(math.sqrt(3)))/2)\*s\*\*2)  
def check(s):  
 if s>0:  
 print("the area of hexagon with side {} is ".format(s),area\_hex(s))  
 else:  
 print("run again and enter a non zero integer value of side ")  
check(side\_hex)

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: a function that returns a base-2 (binary) representation of a base-10 (decimal) string number  
#input  
dec=int(input("enter a decimal no"))  
def dectobin(dec):  
 bin=''  
 while(dec>=1):  
 bin+=str(dec%2)  
 dec=dec//2  
 bin=bin[::-1]  
 return(bin)  
def check(dec):  
 if dec>0:  
 print("the binary representation of decimal {}: ".format(dec),dectobin(dec))  
 else:  
 print("run again and enter a non zero decimal integer")  
check(dec)