Question1

Create a function that takes a string and returns a string in which each character is repeated

once.

Examples

double\_char(&quot;String&quot;) ➞ &quot;SSttrriinngg&quot;

double\_char(&quot;Hello World!&quot;) ➞ &quot;HHeelllloo WWoorrlldd!!&quot;

double\_char(&quot;1234!\_ &quot;) ➞ &quot;11223344!!\_\_ &quot;

A:

ef repeat():

str1 =input(" enter your string")

l=[]

for i in range(len(str1)):

l.append(str1[i]\*2)

print(l)

for i in l:

print(i,end="")

x=repeat()

x

Question2

Create a function that reverses a boolean value and returns the string &quot;boolean expected&quot;

if another variable type is given.

Examples

reverse(True) ➞ False

reverse(False) ➞ True

reverse(0) ➞ &quot;boolean expected&quot;

reverse(None) ➞ &quot;boolean expected&quot;

A:

def reverse(arg=None):

return not arg if type(arg) == bool else "boolean expected"

print(reverse(True)) # False

print(reverse(False)) # True

print(reverse(0)) # "boolean expected"

print(reverse(None)) # "boolean expected"

Question3

Create a function that returns the thickness (in meters) of a piece of paper after folding it n

number of times. The paper starts off with a thickness of 0.5mm.

Examples

num\_layers(1) ➞ &quot;0.001m&quot;

# Paper folded once is 1mm (equal to 0.001m)

num\_layers(4) ➞ &quot;0.008m&quot;

# Paper folded 4 times is 8mm (equal to 0.008m)

num\_layers(21) ➞ &quot;1048.576m&quot;

# Paper folded 21 times is 1048576mm (equal to 1048.576m)

A:

def num\_layers(n):

thickness = 0.5

for \_ in range(n):

thickness \*= 2

return str(thickness / 1000)+'m' # for meters

print(num\_layers(1))

print(num\_layers(4))

print(num\_layers(21))

Question4

Create a function that takes a single string as argument and returns an ordered list containing

the indices of all capital letters in the string.

Examples

index\_of\_caps(&quot;eDaBiT&quot;) ➞ [1, 3, 5]

index\_of\_caps(&quot;eQuINoX&quot;) ➞ [1, 3, 4, 6]

index\_of\_caps(&quot;determine&quot;) ➞ []

A:

def index\_of\_caps():

test\_str = input("enter the string")

print("The original string is : " + str(test\_str))

res = [idx for idx in range(len(test\_str)) if test\_str[idx].isupper()]

print("Uppercase elements indices : " + str(res))

x=index\_of\_caps()

x

index\_of\_caps(&quot;STRIKE&quot;) ➞ [0, 1, 2, 3, 4, 5]

index\_of\_caps(&quot;sUn&quot;) ➞ [1]

Question5

Using list comprehensions, create a function that finds all even numbers from 1 to the given

number.

Examples

find\_even\_nums(8) ➞ [2, 4, 6, 8]

find\_even\_nums(4) ➞ [2, 4]

find\_even\_nums(2) ➞ [2]

A:

class even:

def \_\_init\_\_(self , n ):

self.n = n

def even1(self):

l=[]

for ele in range(self.n+1):

if ele%2!=0 or ele==0 :

continue

else:

l.append(ele)

return l

a=int(input())

obj=even(a)

print(obj.even1())