**Question1**

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

**Examples**

**double\_char("String") ➞ "SSttrriinngg"**

**double\_char("Hello World!") ➞ "HHeelllloo WWoorrlldd!!"**

**double\_char("1234!\_ ") ➞ "11223344!!\_\_ "**

def double\_char(string):

result = ""

for char in string:

result += char \* 2

return result

print(double\_char("String"))

print(double\_char("Hello World!"))

print(double\_char("1234!\_ "))

**Question2**

**Create a function that reverses a boolean value and returns the string "boolean expected" if another variable type is given.**

### Examples

**reverse(True) ➞ False**

**reverse(False) ➞ True**

**reverse(0) ➞ "boolean expected"**

**reverse(None) ➞ "boolean expected"**

def reverse(arg):

if type(arg) == bool:

return not arg

else:

return "boolean expected"

print(reverse(True))

print(reverse(False))

print(reverse(0))

print(reverse(None))

**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) ➞ "0.001m"**

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

**num\_layers(4) ➞ "0.008m"**

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

**num\_layers(21) ➞ "1048.576m"**

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

def num\_layers(n):

thickness = 0.5e-3

for i in range(n):

thickness \*= 2

return str(thickness) + "m"

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("eDaBiT") ➞ [1, 3, 5]**

**index\_of\_caps("eQuINoX") ➞ [1, 3, 4, 6]**

**index\_of\_caps("determine") ➞ []**

**index\_of\_caps("STRIKE") ➞ [0, 1, 2, 3, 4, 5]**

**index\_of\_caps("sUn") ➞ [1]**

def index\_of\_caps(word):

result = []

for i, letter in enumerate(word):

if letter.isupper():

result.append(i)

return result

print(index\_of\_caps("eDaBiT"))

print(index\_of\_caps("eQuINoX"))

print(index\_of\_caps("determine"))

print(index\_of\_caps("STRIKE"))

print(index\_of\_caps("sUn"))

**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]**

def find\_even\_nums(num):

return [x for x in range(2, num + 1, 2)]

print(find\_even\_nums(8))

print(find\_even\_nums(4))

print(find\_even\_nums(2))