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!!\_\_ "

**Ans:**

**def double\_char( str ):**

**tmp = ""**

**for i in str:**

**tmp = tmp + i + i**

**return tmp**

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"

**Ans:**

**def reverse( var):**

**if ( type(var) != type(True)):**

**return "boolean expected"**

**elif(var):**

**return False**

**else:**

**return True**

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)

Ans:

**def num\_layers( n ):**

**for i in range(1 , n +1 ):**

**if ( i == 1):**

**sum = 0.0001**

**else:**

**sum = sum\*2**

**return f"{sum}m"**

#kindly please double check , answer of last test case should be 104.8576m

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]

**Ans:**

**def index\_of\_caps ( str ):**

**lst = []**

**j = 0**

**for i in str:**

**if (ord(i) > 64 and ord(i) < 91):**

**lst.append(j)**

**j = j + 1**

**return lst**

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]

**Ans:**

**def find\_even\_nums( n ):**

**lst =[]**

**for i in range(1 , n + 1 ):**

**if ( i%2 == 0 ):**

**lst.append(i)**

**return lst**