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!!\_\_ "

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

# Declare variables

s1 = "hellohi"

s2 = ""

# print the string variable

print("The given string is :",s1)

# use list comprehension to store string as list

list1 = [c for c in s1]

newList1 = []

# use for loop to iterate list1 and find characters which are repeated atlest twice

for a in list1:

n = list1.count(a)

if n>1:

if newList1.count(a)==0:

newList1.append(a)

# print new list

print(newList1)

# convert list to string

for i in newList1:

s2+= i

# print new string

print(s2)

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"

# 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"

# Define a function , function1

def function1(bool1):

if type(bool1)==bool:

return not bool1

else:

return "Boolean expected"

# check with multiple inputs

input1 = True

input2 = False

input3 = "Text" # some string value

input4 = 0 # some integer value

input5 = None # None value

x = function1(input1)

print(x)

x = function1(input2)

print(x)

x = function1(input3)

print(x)

x = function1(input4)

print(x)

x = function1(input5)

print(x)

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)

# 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.

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

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

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

# Define x in mm

x = 0.5

# Define a function, function1 which returns the thickness of paper in meters

def function1(num):

return (2\*\*num)\*x/1000

# Get the input (no of times paper folded) from the user

n= int(input("Enter number of times paper was folded"))

# call the function by passing n as parameter

x = function1(n)

# print the thickness

print("Paper is ",x,"meters")

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]

# 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.

# declare a string variable

s1 = "hellOhI"

# Declare a function, function1 to get index of upper case characters

def function1(string1):

l1 = []

for i in range(len(s1)):

if s1[i].isupper():

l1.append(i)

return l1

# call the function, function1 and store the return value in list variable, l2

l2 = function1(s1)

# Print the list variable, l2

print(l2)

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]

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

# get the integer number from the user

n = int(input("Enter a number to find all even numbers from zero to the number:"))

# create a list comprehension and store it in a list variable, l2

l2 = [a for a in range(0,n+1) if a%2 ==0]

# print the list containing even numbers

print(l2)