Question1

Create a function that takes a list of strings and integers, and filters out the list so that it returns a list of integers only.

**Examples**

filter\_list([1, 2, 3, "a", "b", 4]) ➞ [1, 2, 3, 4]

filter\_list(["A", 0, "Edabit", 1729, "Python", "1729"]) ➞ [0, 1729]

filter\_list(["Nothing", "here"]) ➞ []

def return\_only\_integer(arr):

  arrR = []

  for i in arr:

    if type(i) == int :

      arrR.append(i)

  return arrR

Question2

Given a list of numbers, create a function which returns the list but with **each element's index in the list added to itself**. This means you add 0 to the number at index 0, add 1 to the number at index 1, etc...

### Examples

add\_indexes([0, 0, 0, 0, 0]) ➞ [0, 1, 2, 3, 4]

add\_indexes([1, 2, 3, 4, 5]) ➞ [1, 3, 5, 7, 9]

add\_indexes([5, 4, 3, 2, 1]) ➞ [5, 5, 5, 5, 5]

def add\_indexes(arr):

  c = 0

  for i in range(len(arr)):

    arr[i] = i+arr[i]

  return arr

Question3

Create a function that takes the height and radius of a cone as arguments and returns the volume of the cone rounded to the nearest hundredth. See the resources tab for the formula.



### Examples

cone\_volume(3, 2) ➞ 12.57

cone\_volume(15, 6) ➞ 565.49

cone\_volume(18, 0) ➞ 0

def cone\_volume(height,radius):

  return 22/7\*radius\*2\*(height/3)

Question4

This Triangular Number Sequence is generated from a pattern of dots that form a triangle. The first 5 numbers of the sequence, or dots, are:

1, 3, 6, 10, 15

This means that the first triangle has just one dot, the second one has three dots, the third one has 6 dots and so on.

Write a function that gives the number of dots with its corresponding triangle number of the sequence.

### Examples

triangle(1) ➞ 1

triangle(6) ➞ 21

triangle(215) ➞ 23220

def triangular( n ):

    j = 1

    k = 1

    for i in range(1, n + 1):

        j = j + 1

        k = k + j

    return k

Question5

Create a function that takes a list of numbers between 1 and 10 (excluding one number) and returns the missing number.

### Examples

missing\_num([1, 2, 3, 4, 6, 7, 8, 9, 10]) ➞ 5

missing\_num([7, 2, 3, 6, 5, 9, 1, 4, 8]) ➞ 10

missing\_num([10, 5, 1, 2, 4, 6, 8, 3, 9]) ➞ 7

def missing\_num(arr):

  mx = max(arr)

  mn = min(arr)

  for i in range(mn,mx):

      if arr.count(i)==0:

        return i