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 filter\_list(n):

l = [i for i in n if isinstance(i, int)] # Check if the item is of integer type

print(l)

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"]) #[]

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(n):

l = []

for i,j in enumerate(n):

res = i + j

l.append(res)

print(l)

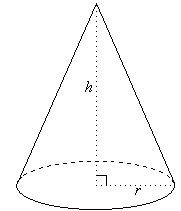
add\_indexes([0, 0, 0, 0, 0])#[0, 1, 2, 3, 4]

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

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

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(h,r):

pi = 3.14

VoC= 1/3\*(pi \* (r\*\*2) \* h)

print(f'{VoC:.2f}')

cone\_volume(3, 2)#12.57

cone\_volume(15, 6)#565.49

cone\_volume(18, 0)#0

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 triangle(n):

# Using the formula for the triangular number: T\_n = n \* (n + 1) // 2

return n \* (n + 1) // 2

# Example usage:

print(triangle(1)) # ➞ 1

print(triangle(6)) # ➞ 21

print(triangle(215)) # ➞ 23220

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(n):

# Define the complete set of numbers from 1 to 10

complete\_set = set(range(1, 11))

# Convert the input list to a set

input\_set = set(n)

# Find the missing numbers by subtracting input set from complete set

missing\_numbers = complete\_set - input\_set

# Print the missing numbers

for missing in missing\_numbers:

print(missing)

# Example usage

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

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

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