**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(lst):

return [x for x in lst if isinstance(x, int)]

# Example Usage

print(filter\_list([1, 2, 3, "a", "b", 4])) # Output: [1, 2, 3, 4]

print(filter\_list(["A", 0, "Edabit", 1729, "Python", "1729"])) # Output: [0, 1729]

print(filter\_list(["Nothing", "here"])) # Output: []

**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(lst):

return [i + num for i, num in enumerate(lst)]

# Example Usage

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

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

print(add\_indexes([5, 4, 3, 2, 1])) # Output: [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**

import math

def cone\_volume(h, r):

volume = (1/3) \* math.pi \* r\*\*2 \* h

return round(volume, 2)

# Example Usage

print(cone\_volume(3, 2)) # Output: 12.57

print(cone\_volume(15, 6)) # Output: 565.49

print(cone\_volume(18, 0)) # Output: 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):

dots = n \* (n + 1) // 2

return dots

# Example Usage

print(triangle(1)) # Output: 1

print(triangle(6)) # Output: 21

print(triangle(215)) # Output: 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(lst):

# Convert the input list to a set

num\_set = set(lst)

# Create a set of numbers from 1 to 10

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

# Find the missing number by subtracting the sets

missing\_number = full\_set - num\_set

# Return the missing number as a single element list

return list(missing\_number)[0]

# Example Usage

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

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

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