**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"]) ➞ []**

**ANS:-**

**lst = [1, 2, 3, 'a', 'b', 4]**

**def filter\_list(lst):**

**intLst = []**

**for i in lst:**

**if type(i) == int:**

**intLst.append(i)**

**return intLst**

**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]**

**ANS:-**

**def add\_indexes(lst):**

**ind = 0**

**index = []**

**for i in lst:**

**index.append(lst.index(i,ind) + i)**

**ind+=1**

**return index**

**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**

**ANS:-**

**import math**

**pi = math.pi**

***# Function to calculate Volume of Cone***

**def cone\_volume(r, h):**

**return round((1 / 3) \* pi \* r \* r \* h)**

***# Driver Code***

**radius = float(5)**

**height = float(12)**

**print( "Volume Of Cone : ", cone\_volume(radius, height) )**

**Volume Of Cone : 314**

**cone\_volume(3, 2)**

**19**

**cone\_volume(15, 6)**

**1414**

**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**

**ANS:-**

**def triangle(n):**

**return n\*(n+1)\*0.5**

**n = int(input('Enter the trinalge number :'))**

**print("The {}th triangle has {} dots ".format(n,int(triangle(n))))**

**Enter the trinalge number :234**

**The 234th triangle has 27495 dots**

**triangle(215)**

**23220.0**

**triangle(1)**

**1.0**

**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**

**ANS:-**

**def missing\_num(lst):**

**total = sum([x for x in range(11)])**

**sum\_Of\_list = sum(lst)**

**return total - sum\_Of\_list**

**print(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**