1. Write a Python program to print "Hello Python"?

|  |
| --- |
| print('Hello World') |

1. Write a Python program to do arithmetical operations addition and division.?

|  |
| --- |
| x= 2  y= 3  print(x+y)  print(y/x) |

1. Write a Python program to find the area of a triangle?

|  |
| --- |
| b= float(input('Base:'))  h= float(input('Height:'))  print(f'Area: {1/2\*b\*h}') |

1. Write a Python program to swap two variables?

|  |
| --- |
| x, y= 2, 4  x, y= y, x  print(x, y) |

1. Write a Python program to generate a random number?

|  |
| --- |
| from random import randint  print(randint(1,10)) |

1. Write a Python program to convert kilometres to miles?

|  |
| --- |
| KM= float(input('kilometers:'))  print(f'miles: {1.609\*KM}') |

1. Write a Python program to convert Celsius to Fahrenheit?

|  |
| --- |
| C= float(input('Celcius:'))  print(f'Fahrenheit: {C\*9/5+32}') |

1. Write a Python program to display calendar?

|  |
| --- |
| import calendar  Y= int(input('Year:'))  M= int(input('Month:'))  print(calendar.month(Y, M)) |

1. Write a Python program to solve quadratic equation?

|  |
| --- |
| import cmath # used for handling complex numbers  a= float(input('Enter x^2 coefficient:'))  b= float(input('Enter x^1 coefficient:'))  c= float(input('Enter x^0 coefficient:'))  discriminant= cmath.sqrt(b\*\*2 - 4\*a\*c)  print(f'root 1: {(-b + discriminant)/(2\*a)}\nroot 2: {(-b - discriminant)/(2\*a)}') |

1. Write a Python program to swap two variables without temp variable?

|  |
| --- |
| x, y= 2, 4  x, y= y, x  print(x, y) |

1. Write a Python Program to Check if a Number is Positive, Negative or Zero?

|  |
| --- |
| n= float(input('number:'))  if n > 0:      print('positive')  elif n < 0:      print('negative')  else:      print('zero') |

1. Write a Python Program to Check if a Number is Odd or Even?

|  |
| --- |
| n= int(input('number:'))  if n%2==0:      print('even')  print('odd') |

1. Write a Python Program to Check Leap Year?

|  |
| --- |
| Y= int(input('Year:'))  if Y%4==0:      print('Leap')  else:      print('Non Leap') |

1. Write a Python Program to Check Prime Number?

|  |
| --- |
| n= int(input('number:'))  for i in range(2,n):      if n%i == 0:          print('not prime')          break      print('prime') |

1. Write a Python Program to Print all Prime Numbers in an Interval of 1-10000?

|  |
| --- |
| def prime(n):      for i in range(2, int(n\*\*0.5)+1):          if n%i == 0:              return False      return True  for num in range(2, 10000+1):      if prime(num):          print(num) |

1. Write a Python Program to Find the Factorial of a Number?

|  |
| --- |
| n= int(input('number:'))  s= 1  for i in range(2, n+1):      s\*=i  print(s) |

1. Write a Python Program to Display the multiplication Table?

|  |
| --- |
| n= int(input('number:'))  for i in range(1, 11):      print(f'{n}x{i}={n\*i}') |

1. Write a Python Program to Print the Fibonacci sequence?

|  |
| --- |
| r= int(input('range:'))  n1= 0  n2= 1  s= None  f= [0, 1]  for i in range(r):      s= n1+n2      n1, n2= n2, s      f.append(s)  print(f) |

1. Write a Python Program to Check Armstrong Number?

|  |
| --- |
| n= input('enter:')  l= len(n)  s= 0  for i in range(l):      s+=int(n[i])\*\*l  print('armstrong' if int(n) == s else 'not armstrong') |

1. Write a Python Program to Find Armstrong Number in an Interval?

|  |
| --- |
| a= int(input('start:'))  b= int(input('end:'))  response= []  for x in range(a, b):      n= str(x)      l= len(n)      s= 0      for i in range(l):          s+=int(n[i])\*\*l      if int(n) == s:          response.append(s)  print(response) |

1. Write a Python Program to Find the Sum of Natural Numbers?

|  |
| --- |
| s= 0  while True:      n= input('enter:')      if n == 'quit':          exit()      s+= int(n)      print(s) |

1. Write a Python Program to Find LCM?

|  |
| --- |
| import math  def LCM(a, b):      return abs(a\*b)//math.gcd(a, b)  # sample  s= LCM(10, 15)  print(s) |

1. Write a Python Program to Find HCF?

|  |
| --- |
| import math  def HCF(a, b):      return math.gcd(a, b)  # sample  s= HCF(5, 35)  print(s) |

1. Write a Python Program to Convert Decimal to Binary, Octal and Hexadecimal?

|  |
| --- |
| n= int(input('number:'))  b= bin(n)  o= oct(n)  h= hex(n)  print(f'binary: {b}\noctadecimal: {o}\nhexadecimal: {h}') |

1. Write a Python Program To Find ASCII value of a character?

|  |
| --- |
| n= input('enter:')  print(ord(n)) |

1. Write a Python Program to Make a Simple Calculator with 4 basic mathematical operations?

|  |
| --- |
| def addition(x, y):      return x+y  def subtraction(x, y):      return x-y  def multiplication(x, y):      return x\*y  def division(x, y):      return x/y  while True:      o= input('enter operation:')      if o == 'quit':          exit()      elif o == '+':          x= float(input('X:'))          y= float(input('Y:'))          print(addition(x, y))      elif o == '-':          x= float(input('X:'))          y= float(input('Y:'))          print(subtraction(x, y))      elif o == '\*':          x= float(input('X:'))          y= float(input('Y:'))          print(multiplication(x, y))      elif o == '/':          x= float(input('X:'))          y= float(input('Y:'))          print(division(x, y)) |

1. Write a Python Program to Display Fibonacci Sequence Using Recursion?

|  |
| --- |
| def Fibonacci(x):      if x <= 0:          return 'enter valid input'      elif x==1:          return 0      elif x==2:          return 1      else:          return Fibonacci(x-1) + Fibonacci(x-2)  for i in range(1, 11):      print(Fibonacci(i)) |

1. Write a Python Program to Find Factorial of Number Using Recursion?

|  |
| --- |
| def factorial(n):      if n == 0 or n == 1:          return 1      else:          return n\*factorial(n-1)  print(factorial(5)) |

1. Write a Python Program to calculate your Body Mass Index?

|  |
| --- |
| def BMI(w, h):      BMI= w/h\*\*2      if BMI < 18.5:          return 'underweight'      elif 18.5 <= BMI <= 24.9:          return 'normal'      elif 25 <= BMI <= 29.9:          return 'overweight'      else:          return 'obese'  print(BMI(85, 1.77)) |

1. Write a Python Program to calculate the natural logarithm of any number?

|  |
| --- |
| import math  def logarithm(n):      return math.log(n)  print(logarithm(10)) |

1. Write a Python Program for cube sum of first n natural numbers?

|  |
| --- |
| def sum\_n\_cube(x):      s= 0      for i in range(1, x+1):          s+= i      return s\*\*3  # sample  print(sum\_n\_cube(3)) |

1. Write a Python Program to find sum of array?

|  |
| --- |
| array= [1, 2, 3, 4, 5, 6, 7, 8, 9]  print(sum(array)) |

1. Write a Python Program to find largest element in an array?

|  |
| --- |
| array= [1, 2, 3, 4, 5, 6, 7, 8, 9]  print(max(array)) |

1. Write a Python Program for array rotation?

|  |
| --- |
| def rotate(array):      new\_array= [array[-1]]      l= len(array)      for i in range(l-1):          new\_array.append(array[i])      return new\_array  # sample  array= [1,2,3,4,5]  print(rotate(rotate(array))) # after 2 rotations |

1. Write a Python Program to Split the array and add the first part to the end?

|  |
| --- |
| array= [1, 2, 3, 4, 5, 6, 7, 8, 9]  l= len(array)  m= l//2 if l%2==0 else (l+1)//2  a1, a2= array[:m], array[m:]  new\_array= a2+a1  print(new\_array) |

1. Write a Python Program to check if given array is Monotonic?

|  |
| --- |
| a= [1, 2, 3, 4, 5, 6]  b= [6, 5, 4, 3, 2, 1]  c= [1, 1, 1, 1, 1, 1]  d= [1, 2, 4, 2, 5, 2]  def check\_monotonic(array):      response= []      l= len(array)      for i in range(1, l):          if array[i]-array[i-1] > 0:              response.append('increasing')          elif array[i]-array[i-1] < 0:              response.append('decreasing')          else:              response.append('no change')      if len(set(response)) == 1:          return 'monotonic'      return 'not monotonic'  # sample  print(check\_monotonic(a))  print(check\_monotonic(b))  print(check\_monotonic(c))  print(check\_monotonic(d)) |

1. Write a Python Program to Add Two Matrices?

|  |
| --- |
| m1= [[1, 2],       [3, 4]]  m2= [[3, 6],       [7, 9]]  print(f'[{m1[0][0]+m2[0][0]} {m1[0][1]+m2[0][1]}\n{m1[1][0]+m2[1][0]} {m1[1][1]+m2[1][1]}]') |

1. Write a Python Program to Multiply Two Matrices?

|  |
| --- |
| m1= [[1, 2],       [3, 4]]  m2= [[3, 6],       [7, 9]]  s00= m1[0][0]\*m2[0][0]+m1[0][1]\*m2[1][0]  s01= m1[0][0]\*m2[0][1]+m1[0][1]\*m2[1][1]  s10= m1[1][0]\*m2[0][0]+m1[1][1]\*m2[1][0]  s11= m1[1][0]\*m2[0][1]+m1[1][1]\*m2[1][1]  print(f'{s00} {s01}\n{s10} {s11}') |

1. Write a Python Program to Transpose a Matrix?

|  |
| --- |
| matrix= [[1, 2],           [3, 4],           [5, 6]]  transpose= map(list, zip(\*matrix))  for i in transpose:      print(i) |

1. Write a Python Program to Sort Words in Alphabetic Order?

|  |
| --- |
| word\_bag= ['apple', 'dragon', 'apex', 'lemon']  word\_bag.sort()  print(word\_bag) |

1. Write a Python Program to Remove Punctuation From a String?

|  |
| --- |
| punctuations="""'"!@#$%^&\*()\_+-=/;:.,[{]}"""  string= 'This morning, I was walking my dog!'  no\_punc= ''  for i in (string):      if i in punctuations:          continue      no\_punc+=i  print(no\_punc) |

1. Write a Python program to check if the given number is a Disarium Number?

|  |
| --- |
| n= input('number:')  l= len(n)  s= 0  for i in range(1, l+1):      s+= int(n[i-1])\*\*i  if int(n) == s:      print('disarium number')  else:      print('some vanilla ass number.') |

1. Write a Python program to print all disarium numbers between 1 to 100?

|  |
| --- |
| disarium= []  for n in range(1, 101):      n=  str(n)      l= len(n)      s= 0      for i in range(1, l+1):          s+= int(n[i-1])\*\*i      if int(n) == s:          disarium.append(s)  print(disarium) |

1. Write a Python program to check if the given number is Happy Number?

|  |
| --- |
| def happy\_number(n):      n= str(n)      l= len(n)      s= 0      for i in range(l):          s+= int(n[i])\*\*2      return s  n= 81 # sample number  i= 0 # iteration count  while True:      i+=1      print(f'iteration: {i}\nvalue: {n}\n')      if n == 1:          print('happy number')          exit()      if i == 1000:          print(f'iteration limit 1000 reached.\nnot a happy number.')          exit()      f= happy\_number(n)      n= f |

1. Write a Python program to print all happy numbers between 1 and 100?

|  |
| --- |
| # function for iteration  def happy\_number\_iteration(n):      s= 0      n= str(n)      l= len(n)      for i in range(l):          s+= int(n[i])\*\*2      return s  # function for checking if a number is a happy number or not  def check\_happy\_number(n):      for i in range(1000):          f= n          s= happy\_number\_iteration(f)          n= s          if n == 1:              return n  # list comprehension  happy\_numbers= [i for i in range(1, 101) if check\_happy\_number(i) == 1]  print(happy\_numbers) |

1. Write a Python program to determine whether the given number is a Harshad Number?

|  |
| --- |
| def harshad\_number(n):      s= 0      str\_n= str(n)      l= len(str\_n)      for i in range(l):          s+= int(str\_n[i])      if n%s == 0:          return 'harshad number'      else:          return 'some random ass number'  # sample  print(harshad\_number(18))  print(harshad\_number(15)) |

1. Write a Python program to print all pronic numbers between 1 and 100?

|  |
| --- |
| harshad\_numbers= []  for n in range(1, 101):      s= 0      str\_n= str(n)      l= len(str\_n)      for i in range(l):          s+= int(str\_n[i])      if n%s == 0:          harshad\_numbers.append(n)  # result  print(harshad\_numbers) |

1. Write a Python program to find sum of elements in list?

|  |
| --- |
| l= [1, 2, 3, 4, 5]  print(sum(l)) |

1. Write a Python program to Multiply all numbers in the list?

|  |
| --- |
| from functools import reduce  l= [1, 2, 3, 4, 5]  print(reduce(lambda x, y: x\*y, l)) |

1. Write a Python program to find smallest number in a list?

|  |
| --- |
| l= [1, 2, 3, 4, 5]  s= sum(l)  for i in l:      if i < s:          s = i  print(s) |

1. Write a Python program to find largest number in a list?

|  |
| --- |
| l= [1, 2, 3, 4, 5]  s= 0  for i in l:      if i > s:          s = i  print(s) |

1. Write a Python program to find second largest number in a list?

|  |
| --- |
| l= [1, 4, 2, 3, 5]  l.sort()  print(l[-2]) |

1. Write a Python program to find N largest elements from a list?

|  |
| --- |
| n= int(input('enter Nth number:'))  l= [1, 4, 2, 3, 5]  l.sort()  print(l[-n:]) |

1. Write a Python program to print even numbers in a list?

|  |
| --- |
| l= [1, 4, 2, 3, 5]  e= [i for i in l if i%2==0]  print(e) |

1. Write a Python program to print odd numbers in a List?

|  |
| --- |
| l= [1, 4, 2, 3, 5]  o= [i for i in l if i%2!=0]  print(o) |

1. Write a Python program to Remove empty List from List?

|  |
| --- |
| l= [1, 2, 'aaa', None, 5, None]  remove\_empty= []  for i in l:      if i == None:          continue      else:          remove\_empty.append(i)  print(remove\_empty) |

1. Write a Python program to Cloning or Copying a list?

|  |
| --- |
| l= [1, 2, 3, 4, 5]  c= l.copy()  print(c) |

1. Write a Python program to Count occurrences of an element in a list?

|  |
| --- |
| l= [1, 2, 2, 3, 2, 4, 5, 2]  print(l.count(2)) |

1. Write a Python program to find words which are greater than given length k?

|  |
| --- |
| k= int(input('enter threshold length:'))  l= ['ant', 'dog', 'rooster', 'dinosaur', 'dragon', 'lion']  n= []  for i in l:      if len(i) > k:          n.append(i)  print(n) |

1. Write a Python program for removing i-th character from a string?

|  |
| --- |
| s= 'hellooooo... kya karre hoooo...'  n= ''  i= int(input('enter the character index:'))  if 0 > i > len(s):      print('out of bound')      exit()  for index, value in enumerate(s):      if index == i:          continue      else:          n+=value  print(n) |

1. Write a Python program to split and join a string?

|  |
| --- |
| st= 'hello doctor strange.'  s= st.split()  j= '-'.join(s)  print(j) |

1. Write a Python to check if a given string is binary string or not?

|  |
| --- |
| w\_s= 122122212  b\_s= 110010001  b\_s\_0= 0000000000  def binary\_string\_check(string):      string= str(string)      s= set(string)      if len(s) == 2 and ('0' in s and '1' in s) or (len(s) == 1 and ('0' in s or '1' in s)):          return True      return False  print(binary\_string\_check(w\_s))  print(binary\_string\_check(b\_s))  print(binary\_string\_check(b\_s\_0)) |

1. Write a Python program to find uncommon words from two Strings?

|  |
| --- |
| s1= 'hello mister Parker, are you Spider-Man ?'  s2= 'hello miss Jane, are you Cat Woman ?'  s1\_l= s1.split()  s2\_l= s2.split()  s1\_l= list(set(s1\_l))  s2\_l= list(set(s2\_l))  unique= []  for i in s1\_l:      if i not in s2\_l:          unique.append(i)  for i in s2\_l:      if i not in s1\_l:          unique.append(i)  print(unique) |

1. Write a Python to find all duplicate characters in string?

|  |
| --- |
| s= 'apples'  set\_s= set(s)  d= {}  for i in s:      if i in d:          d[i]+=1      else:          d[i]=1  duplicates= []  for j in set\_s:      if d[j]>1:          duplicates.append(j)      else:          continue  print(duplicates) |

1. Write a Python Program to check if a string contains any special character?

|  |
| --- |
| special\_chars= '!@#$%^&\*+-\_'  s= 'a%ws we are!'  for i in s:      if i in special\_chars:          print('has special characters.')          break |

1. Write a Python program to Extract Unique values dictionary values?

|  |
| --- |
| d= {'a': 12,      't': 14,      14 : 12,      'r': 14}  values= list(d.values())  print(list(set(values))) |

1. Write a Python program to find the sum of all items in a dictionary?

|  |
| --- |
| d= {1: 5,      2: 7,      'k': 6,      'g': 8,      0: 'er'}  key\_list= list(d.keys())  s= 0  for i in key\_list:      if type(d[i]) == int:          s+=d[i]      else:          continue  print(s) |

1. Write a Python program to Merging two Dictionaries?

|  |
| --- |
| d1= {'a': 1,       'b': 2}  d2= {'c': 3,       'd': 4}  d1.update(d2)  print(d1) |

1. Write a Python program to convert key-values list to flat dictionary?

|  |
| --- |
| d= {'key':['a', 'b', 'c'],      'value':[1, 2, 3]}  flatten= dict(zip(d['key'], d['value']))  print(flatten) |

1. Write a Python program to insertion at the beginning in OrderedDict?

|  |
| --- |
| from collections import OrderedDict  def beg\_ins(d, k, v):      d.update({k: v})      d.move\_to\_end(k, last= False)      return d  d= OrderedDict([('a', 1),                  ('b', 2),                  ('c', 3)])  print(beg\_ins(d, 'd', 4)) |

1. Write a Python program to check order of character in string using OrderedDict()?

|  |
| --- |
|  |

1. Write a Python program to sort Python Dictionaries by Key or Value?

|  |
| --- |
| d= {'a': 1,      's': 9,      'c': 7,      'e': 10}  print(sorted(d.items(), key= lambda x: x[0])) # sort by keys  print(sorted(d.items(), key= lambda x: x[1])) # sort by values |

|  |
| --- |
| Question 1: |
|  |

|  |
| --- |
| Write a program that calculates and prints the value according to the given formula: |
|  |

|  |
| --- |
| Q = Square root of [(2 \* C \* D)/H] |
|  |

|  |
| --- |
| Following are the fixed values of C and H: |
|  |

|  |
| --- |
| C is 50. H is 30. |
|  |

|  |
| --- |
| D is the variable whose values should be input to your program in a comma-separated sequence. |
|  |

|  |
| --- |
| Example |
|  |

|  |
| --- |
| Let us assume the following comma separated input sequence is given to the program: |
|  |

|  |
| --- |
| 100,150,180 |
|  |

|  |
| --- |
| The output of the program should be: |
|  |

18,22,24

|  |
| --- |
| from math import sqrt  def calc\_Q(D, C= 50, H= 30):      return sqrt((2\*C\*D)/H)  D= [100, 150, 180]  result= list(map(calc\_Q, D))  print(result) |

|  |  |
| --- | --- |
| Question 2: |  |
| Write a program which takes 2 digits, X,Y as input and generates a 2-dimensional array. The element value in the i-th row and j-th column of the array should be i\*j. | |
|  | |

|  |
| --- |
| Note: i=0,1.., X-1; j=0,1,...Y-1 |
|  |

|  |
| --- |
| Example |
|  |

|  |
| --- |
| Suppose the following inputs are given to the program: |
|  |

|  |
| --- |
| 3,5 |
|  |

|  |
| --- |
| Then, the output of the program should be: |
|  |

|  |
| --- |
| [[0, 0, 0, 0, 0], [0, 1, 2, 3, 4], [0, 2, 4, 6, 8]] |
|  |
| X= 3  Y= 5  matrix= []  for i in range(X):      row= []      for j in range(Y):          row.append(i\*j)      matrix.append(row)  print(matrix) |

Question 3:

|  |
| --- |
| Write a program that accepts a comma separated sequence of words as input and prints the words in a comma-separated sequence after sorting them alphabetically. |
|  |

|  |
| --- |
| Suppose the following input is supplied to the program: |
|  |

|  |
| --- |
| without,hello,bag,world |
|  |

|  |
| --- |
| Then, the output should be: |
|  |

bag,hello,without,world

|  |
| --- |
| def sort\_words(\*args):      sorted\_words= sorted(args)      return(', '.join(sorted\_words))  print(sort\_words('without','hello','bag','world')) |

Question 4:

|  |
| --- |
| Write a program that accepts a sequence of whitespace separated words as input and prints the words after removing all duplicate words and sorting them alphanumerically. |
|  |

|  |
| --- |
| Suppose the following input is supplied to the program: |
|  |

|  |
| --- |
| hello world and practice makes perfect and hello world again |
|  |

|  |
| --- |
| Then, the output should be: |
|  |

again and hello makes perfect practice world

|  |
| --- |
| def pure\_text(string):      chars= string.split()      set\_chars= list(set(chars))      set\_chars.sort()      return ' '.join(set\_chars)  print(pure\_text('hello world and practice makes perfect and hello world again')) |

Question 5:

|  |
| --- |
| Write a program that accepts a sentence and calculate the number of letters and digits. |
|  |

|  |
| --- |
| Suppose the following input is supplied to the program: |
|  |

|  |
| --- |
| hello world! 123 |
|  |

|  |
| --- |
| Then, the output should be: |
|  |

|  |
| --- |
| LETTERS 10 |
|  |

DIGITS 3

|  |
| --- |
| def count\_datatype(string):      a= 0      n= 0      for char in string:          if char.isalpha():              a+=1          elif char.isdigit():              n+=1          else:              continue      return f'alphabets: {a}\ndigits: {n}'  # sample  print(count\_datatype('hello world! 123')) |

Question 6:

|  |
| --- |
| A website requires the users to input username and password to register. Write a program to check the validity of password input by users. |
|  |

|  |
| --- |
| Following are the criteria for checking the password: |
|  |

|  |
| --- |
| 1. At least 1 letter between [a-z] |
|  |

|  |
| --- |
| 2. At least 1 number between [0-9] |
|  |

|  |
| --- |
| 1. At least 1 letter between [A-Z] |
|  |

|  |
| --- |
| 3. At least 1 character from [$#@] |
|  |

|  |
| --- |
| 4. Minimum length of transaction password: 6 |
|  |

|  |
| --- |
| 5. Maximum length of transaction password: 12 |
|  |

|  |
| --- |
| Your program should accept a sequence of comma separated passwords and will check them according to the above criteria. Passwords that match the criteria are to be printed, each separated by a comma. |
|  |

|  |
| --- |
| Example |
|  |

|  |
| --- |
| If the following passwords are given as input to the program: |
|  |

|  |
| --- |
| ABd1234@1,a F1#,2w3E\*,2We3345 |
|  |

|  |
| --- |
| Then, the output of the program should be: |
|  |

ABd1234@1

|  |
| --- |
| def check\_password(\*args):      result = []      for password in args:          length  = 6 < len(password) < 12          digit   = any(char.isdigit() for char in password)          cap\_a   = any(char.isupper() for char in password)          low\_a   = any(char.islower() for char in password)          special = any(char in '$#@' for char in password)          if length and digit and cap\_a and low\_a and special:              result.append(password)      return result  # sample  valid\_passwords = check\_password('ABd1234@1', 'a F1#', '2w3E\*', '2We3345')  print(valid\_passwords) |

|  |
| --- |
| Question 1: |
|  |

Define a class with a generator which can iterate the numbers, which are divisible by 7, between a given range 0 and n.

|  |
| --- |
| # copied from chatGPT  class DivisibilityTest:      def \_\_init\_\_(self, n):          self.n= n      def test(self):          for i in range(self.n+1):              if i%7 == 0:                  yield i  # sample  n= 50  generator\_instance= DivisibilityTest(n)  for n in generator\_instance.test():      print(n) |

Question 2:

|  |
| --- |
| Write a program to compute the frequency of the words from the input. The output should output after sorting the key alphanumerically. |
|  |

|  |
| --- |
| Suppose the following input is supplied to the program: |
|  |

|  |
| --- |
| New to Python or choosing between Python 2 and Python 3? Read Python 2 or Python 3. |
|  |

|  |
| --- |
| Then, the output should be: |
|  |

|  |
| --- |
| 2:2 |
|  |

|  |
| --- |
| 3.:1 |
|  |

|  |
| --- |
| 3?:1 |
|  |

|  |
| --- |
| New:1 |
|  |

|  |
| --- |
| Python:5 |
|  |

|  |
| --- |
| Read:1 |
|  |

|  |
| --- |
| and:1 |
|  |

|  |
| --- |
| between:1 |
|  |

|  |
| --- |
| choosing:1 |
|  |

|  |
| --- |
| or:2 |
|  |

to:1

|  |
| --- |
| def check\_frequency(string):      l= string.split()      l.sort()      d= {}      for i in l:          if i in d:              d[i]+= 1          else:              d[i]= 1      return d  print(check\_frequency('New to Python or choosing between Python 2 and Python 3? Read Python 2 or Python 3.')) |

|  |
| --- |
| Question 3: |
|  |

|  |
| --- |
|  |
|  |

Define a class Person and its two child classes: Male and Female. All classes have a method "getGender" which can print "Male" for Male class and "Female" for Female class.

|  |
| --- |
| class Person:      def getGender(self):          return 'Unknown'  class Male(Person):      def getGender(self):          return 'Male'  class Female(Person):      def getGender(self):          return 'Female'  # sample  RamLal= Male()  print(RamLal.getGender()) |

Question 4:

Please write a program to generate all sentences where subject is in ["I", "You"] and verb is in ["Play", "Love"] and the object is in ["Hockey","Football"].

|  |
| --- |
| subj= ['I', 'You']  verb= ['Play', 'Love']  obje= ['Hockey', 'Football']  for i in subj:      for j in verb:          for k in obje:              print(i+' '+j+' '+k+'.') |

Question 5:

Please write a program to compress and decompress the string "hello world!hello world!hello world!hello world!".

|  |
| --- |
| # copied from chatGPT  import zlib  def compress(string):      compressed\_data= zlib.compress(string.encode('utf-8'))      return compressed\_data  def decompress(compressed\_data):      decompressed\_data= zlib.decompress(compressed\_data).decode('utf-8')      return decompressed\_data  # sample  string = "hello world!hello world!hello world!hello world!"  # Compress the string  compressed\_data = compress(string)  print("Compressed:", compressed\_data)  # Decompress the string  decompressed\_string = decompress(compressed\_data)  print("Decompressed:", decompressed\_string) |

Question 6:

Please write a binary search function which searches an item in a sorted list. The function should return the index of element to be searched in the list.

|  |
| --- |
| def binary\_search(array, target):      # basic parameters      lower\_index= 0      upper\_index= len(array)-1      # iterative assessment      while lower\_index <= upper\_index:          middle\_index= (upper\_index+lower\_index)//2          if array[middle\_index] < target:              lower\_index= middle\_index + 1          elif array[middle\_index] > target:              upper\_index= middle\_index - 1          else:              return middle\_index      return -1  # sample  array = [2, 5, 6, 1, 7, 9]  result= binary\_search(array, 7)  # display  print('target not found' if result == -1 else result) |

Question 1:

Please write a program using generator to print the numbers which can be divisible by 5 and 7 between 0 and n in comma separated form while n is input by console.

Example:  
If the following n is given as input to the program:

100

Then, the output of the program should be:

0,35,70

|  |
| --- |
| # generator function  def generator(n):      for i in range(n+1):          yield i  # testing function  def divisibility(n):      result= []      for value in generator(n):          if value%5 == 0 and value%7 == 0:              result.append(value)      return ', '.join(map(str, result))  # sample  result= divisibility(100)  print(result) |

Question 2:

Please write a program using generator to print the even numbers between 0 and n in comma separated form while n is input by console.

Example:  
If the following n is given as input to the program:

10

Then, the output of the program should be:

0,2,4,6,8,10

|  |
| --- |
| # generator  def generator(n):      for i in range(n+1):          yield i  # method  def filter\_even(n):      result= []      for value in generator(n):          if value%2 == 0:              result.append(value)      return ', '.join(map(str, result))  # sample  print(filter\_even(50)) |

Question 3:

The Fibonacci Sequence is computed based on the following formula:

f(n)=0 if n=0  
f(n)=1 if n=1  
f(n)=f(n-1)+f(n-2) if n>1

Please write a program using list comprehension to print the Fibonacci Sequence in comma separated form with a given n input by console.

Example:  
If the following n is given as input to the program:

7

Then, the output of the program should be:

0,1,1,2,3,5,8,13

|  |
| --- |
| def print\_fibonacci(n):      series= [0, 1]      series.extend(series[i-1]+series[i-2] for i in range(2, n))      if n == 1:          return series[0]      elif n > 2:          return ', '.join(map(str, series))  print(print\_fibonacci(7)) |

Question 4:

Assuming that we have some email addresses in the "[username@companyname.com](mailto:username@companyname.com)" format, please write program to print the user name of a given email address. Both user names and company names are composed of letters only.

Example:  
If the following email address is given as input to the program:

[john@google.com](mailto:john@google.com)

Then, the output of the program should be:

john

|  |
| --- |
| def fetch\_info(emailID):      name= emailID.split('@')[0]      company= (emailID.split('@')[1]).split('.')[0]      return f'Name: {name}\nCompany: {company}'  # sample  print(fetch\_info('john@google.com')) |

Question 5:

Define a class named Shape and its subclass Square. The Square class has an init function which takes a length as argument. Both classes have a area function which can print the area of the shape where Shape's area is 0 by default.

|  |
| --- |
| class Shape:      def area(self):          return 0  class Square(Shape):      def \_\_init\_\_(self, length):          self.length= length      def area(self):          return self.length\*\*2  # sample  s= Square(5)  print(s.area()) |

Question1. Write a function that stutters a word as if someone is struggling to read it. The first two letters are repeated twice with an ellipsis ... and space after each, and then the word is pronounced with a question mark ?.

**Examples**

stutter("incredible") ➞ "in... in... incredible?"

stutter("enthusiastic") ➞ "en... en... enthusiastic?"

stutter("outstanding") ➞ "ou... ou... outstanding?"

Hint :- Assume all input is in lower case and at least two characters long.

|  |
| --- |
| def stutter(word):      output= ''      for i in range(3):          if i < 2:              output+= word[0]+word[1]+'... '          else:              output+=word+'?'      return output  print(stutter('hello')) |

Question 2.Create a function that takes an angle in radians and returns the corresponding angle in degrees rounded to one decimal place.

### Examples

radians\_to\_degrees(1) ➞ 57.3

radians\_to\_degrees(20) ➞ 1145.9

radians\_to\_degrees(50) ➞ 2864.8

|  |
| --- |
| import math  def rad\_to\_deg(radian):      return round(radian\*(180/math.pi), 2)  # sample  print(rad\_to\_deg(1)) |

Question 3. In this challenge, establish if a given integer num is a Curzon number. If 1 plus 2 elevated to num is exactly divisible by 1 plus 2 multiplied by num, then num is a Curzon number.

Given a non-negative integer num, implement a function that returns True if num is a Curzon number, or False otherwise.

### Examples

is\_curzon(5) ➞ True

# 2 \*\* 5 + 1 = 33

# 2 \* 5 + 1 = 11

# 33 is a multiple of 11

is\_curzon(10) ➞ False

# 2 \*\* 10 + 1 = 1025

# 2 \* 10 + 1 = 21

# 1025 is not a multiple of 21

is\_curzon(14) ➞ True

# 2 \*\* 14 + 1 = 16385

# 2 \* 14 + 1 = 29

# 16385 is a multiple of 29

|  |
| --- |
| def is\_curzon(n):      if (2\*\*n+1)%(2\*n+1) == 0:          return True      return False  # sample  print(is\_curzon(5))  print(is\_curzon(10)) |

Question 4.Given the side length x find the area of a hexagon.



### Examples

area\_of\_hexagon(1) ➞ 2.6

area\_of\_hexagon(2) ➞ 10.4

area\_of\_hexagon(3) ➞ 23.4

|  |
| --- |
| def hex\_area(s):      return round((3\*3\*\*(1/2)\*s\*\*2)/(2), 1)  # sample  print(hex\_area(1)) |

Question 5. Create a function that returns a base-2 (binary) representation of a base-10 (decimal) string number. To convert is simple: ((2) means base-2 and (10) means base-10) 010101001(2) = 1 + 8 + 32 + 128.

Going from right to left, the value of the most right bit is 1, now from that every bit to the left will be x2 the value, value of an 8 bit binary numbers are (256, 128, 64, 32, 16, 8, 4, 2, 1).

### Examples

binary(1) ➞ "1"

# 1\*1 = 1

binary(5) ➞ "101"

# 1\*1 + 1\*4 = 5

binary(10) ➞ "1010"

# 1\*2 + 1\*8 = 10

|  |
| --- |
| def bin\_to\_dec(b):      if type(b) == int:          b= str(b)      s= 0      for i, v in enumerate(b):          s+= int(v)\*2\*\*i      return s  # sample  print(bin\_to\_dec(101)) |

Question1. Create a function that takes three arguments a, b, c and returns the sum of the numbers that are evenly divided by c from the range a, b inclusive.

**Examples**

evenly\_divisible(1, 10, 20) ➞ 0

# No number between 1 and 10 can be evenly divided by 20.

evenly\_divisible(1, 10, 2) ➞ 30

# 2 + 4 + 6 + 8 + 10 = 30

evenly\_divisible(1, 10, 3) ➞ 18

# 3 + 6 + 9 = 18

|  |
| --- |
| def evenly\_divide(a, b, c):      n= [i for i in range(a, b+1)]      r= [j for j in n if j%c == 0]      return sum(r)  # sample  print(evenly\_divide(1, 10, 2)) |

Question2. Create a function that returns True if a given inequality expression is correct and False otherwise.

### Examples

correct\_signs("3 < 7 < 11") ➞ True

correct\_signs("13 > 44 > 33 > 1") ➞ False

correct\_signs("1 < 2 < 6 < 9 > 3") ➞ True

|  |
| --- |
| def evaluate(expression):      return eval(expression)  # sample  print(evaluate('3 < 7 < 11')) |

Question3. Create a function that replaces all the vowels in a string with a specified character.

### Examples

replace\_vowels("the aardvark", "#") ➞ "th# ##rdv#rk"

replace\_vowels("minnie mouse", "?") ➞ "m?nn?? m??s?"

replace\_vowels("shakespeare", "\*") ➞ "sh\*k\*sp\*\*r\*"

Question4. Write a function that calculates the **factorial** of a number **recursively**.

### Examples

factorial(5) ➞ 120

factorial(3) ➞ 6

factorial(1) ➞ 1

factorial(0) ➞ 1

|  |
| --- |
| def replace\_vowels(string, character):      vowels= 'aeiou'      output=''      for i in string:          if i in vowels:              output+= character          else:              output+= i      return output  # sample  print(replace\_vowels("the aardvark", "#")) |

**Question 5**

**Hamming distance** is the number of characters that differ between two strings.

To illustrate:

String1: "abcbba"

String2: "abcbda"

Hamming Distance: 1 - "b" vs. "d" is the only difference.

Create a function that computes the **hamming distance** between two strings.

### Examples

hamming\_distance("abcde", "bcdef") ➞ 5

hamming\_distance("abcde", "abcde") ➞ 0

hamming\_distance("strong", "strung") ➞ 1

|  |
| --- |
| # copied from chatGPT  def hamming\_distance(str1, str2):      if len(str1) != len(str2):          raise ValueError('Input strings must have the same length')      distance= sum(bit1 != bit2 for bit1, bit2 in zip(str1, str2))      return distance  # sample  str1= 'kratos'  str2= 'pratoz'  print(hamming\_distance(str1, str2)) |

Question 1

Create a function that takes a list of non-negative **integers** and **strings** and return a new list without the strings.

**Examples**

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

filter\_list([1, "a", "b", 0, 15]) ➞ [1, 0, 15]

filter\_list([1, 2, "aasf", "1", "123", 123]) ➞ [1, 2, 123]

|  |
| --- |
| l= [1, 2, 'a', 'b']  s= [i for i in l if type(i) == int]  print(s) |

Question 2

The "Reverser" takes a string as input and returns that string in reverse order, with the opposite case.

### Examples

reverse("Hello World") ➞ "DLROw OLLEh"

reverse("ReVeRsE") ➞ "eSrEvEr"

reverse("Radar") ➞ "RADAr"

|  |
| --- |
| def reverse(string):      s=''      for i in string:          if i.islower():              s+=i.upper()          else:              s+=i.lower()      return s[::-1]  #sample  s= 'Radar'  print(reverse(s)) |

Question 3

You can assign variables from lists like this:

lst = [1, 2, 3, 4, 5, 6]

first = lst[0]

middle = lst[1:-1]

last = lst[-1]

print(first) ➞ outputs 1

print(middle) ➞ outputs [2, 3, 4, 5]

print(last) ➞ outputs 6

With Python 3, you can assign variables from lists in a much more succinct way. Create variables first, middle and last from the given list using **destructuring assignment** (check the **Resources** tab for some examples), where:

first ➞ 1

middle ➞ [2, 3, 4, 5]

last ➞ 6

Your task is to unpack the list writeyourcodehere into three variables, being first, middle, and last, with middle being everything in between the first and last element. Then print all three variables.

|  |
| --- |
| def split\_list(l):      fst= l[0]      lst= l[-1]      mid= l[1:-1]      return f'first: {fst}\nmiddle: {mid}\nlast: {lst}'  #sample  print(split\_list([1,2,3,4,5])) |

Question 4

Write a function that calculates the **factorial** of a number **recursively**.

### Examples

factorial(5) ➞ 120

factorial(3) ➞ 6

factorial(1) ➞ 1

factorial(0) ➞ 1

|  |
| --- |
| def fact(n):      if n == 0 or n == 1:          return 1      else:          return n\*fact(n-1)  # sample  print(fact(5)) |

Question 5

Write a function that moves all elements of one type to the **end** of the list.

### Examples

move\_to\_end([1, 3, 2, 4, 4, 1], 1) ➞ [3, 2, 4, 4, 1, 1]

# Move all the 1s to the end of the array.

move\_to\_end([7, 8, 9, 1, 2, 3, 4], 9) ➞ [7, 8, 1, 2, 3, 4, 9]

move\_to\_end(["a", "a", "a", "b"], "a") ➞ ["b", "a", "a", "a"]

|  |
| --- |
| def move\_to\_end(array, char):      for i in range(len(array)):          if array[i] == char:              sub= array.pop(i)              array.append(sub)      return array  # sample  d= [1, 2, 3, 2, 4]  c= 2  print(move\_to\_end(d, c)) |

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!!\_\_ "

|  |
| --- |
| def drunken\_master(string):      s= ''      for i in string:          for \_ in range(2):              s+=i      return s  # sample  print(drunken\_master('Hello')) |

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"

|  |
| --- |
| def reverse(input):      if type(input) == bool:          return not(input)      else:          return 'boolean expected'  # sample  print(reverse(True)) |

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)

|  |
| --- |
| def fold(n):      paper\_thickness= 0.5/1000      return 2\*\*n\*paper\_thickness  # sample  print(fold(21)) |

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]

|  |
| --- |
| def index\_of\_caps(string):      index= []      for i, v in enumerate(string):          if v.isupper():              index.append(i)      return index  # sample  print(index\_of\_caps('sUn')) |

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]

|  |
| --- |
| def find\_even\_nums(n):      return [i for i in range(1, n) if i%2==0]  # sample  print(find\_even\_nums(10)) |

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"]) ➞ []

|  |
| --- |
| l= [1, 2, 'a', 'b']  s= [i for i in l if type(i) == int]  print(s) |

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(l):      result= []      for i, v in enumerate(l):          result.append(i+v)      return result  # sample  print(add\_indexes([1, 2, 3, 4, 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):      return round((1/3)\*math.pi\*r\*\*2\*h, 2)  # sample  print(cone\_volume(3, 2)) |

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):      seq= [1]      seq.extend(seq[i-1]+i+1 for i in range(1, n))      return seq[-1]  # sample  print(triangle(1))  print(triangle(2))  print(triangle(6)) |

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(array):      seq= [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]      for i in seq:          if i not in array:              return i  # sample  print(missing\_num([1, 2, 3, 4, 6, 7, 8, 9, 10])) |

Question1

Write a function that takes a list and a number as arguments. Add the number to the end of the list, then remove the first element of the list. The function should then return the updated list.

**Examples**

next\_in\_line([5, 6, 7, 8, 9], 1) ➞ [6, 7, 8, 9, 1]

next\_in\_line([7, 6, 3, 23, 17], 10) ➞ [6, 3, 23, 17, 10]

next\_in\_line([1, 10, 20, 42 ], 6) ➞ [10, 20, 42, 6]

next\_in\_line([], 6) ➞ "No list has been selected"

|  |
| --- |
| def next\_in\_line(array, next):      if array == []:          print('no list has been selected')          exit()      array.pop(0)      array.append(next)      return array  # sample  print(next\_in\_line([5, 6, 7, 8, 9], 1))  print(next\_in\_line([], 6)) |

Question2

Create the function that takes a list of dictionaries and returns the sum of people's budgets.

### Examples

get\_budgets([

{ "name": "John", "age": 21, "budget": 23000 },

{ "name": "Steve", "age": 32, "budget": 40000 },

{ "name": "Martin", "age": 16, "budget": 2700 }

]) ➞ 65700

get\_budgets([

{ "name": "John", "age": 21, "budget": 29000 },

{ "name": "Steve", "age": 32, "budget": 32000 },

{ "name": "Martin", "age": 16, "budget": 1600 }

]) ➞ 62600

|  |
| --- |
| def get\_budgets(list\_of\_dict):      s= 0      for entry in list\_of\_dict:          s+= entry['budget']      return s  # sample  print(get\_budgets([{ "name": "John", "age": 21, "budget": 23000 },                     { "name": "Steve",  "age": 32, "budget": 40000 },                     { "name": "Martin",  "age": 16, "budget": 2700 }])) |

Question3

Create a function that takes a string and returns a string with its letters in alphabetical order.

### Examples

alphabet\_soup("hello") ➞ "ehllo"

alphabet\_soup("edabit") ➞ "abdeit"

alphabet\_soup("hacker") ➞ "acehkr"

alphabet\_soup("geek") ➞ "eegk"

alphabet\_soup("javascript") ➞ "aacijprstv"

|  |
| --- |
| def alphabet\_soup(string):      char\_list= []      for i in string:          char\_list.append(i)      char\_list.sort()      return ''.join(char\_list)  # sample  print(alphabet\_soup("hello")) |

Question4

Suppose that you invest $10,000 for 10 years at an interest rate of 6% compounded monthly. What will be the value of your investment at the end of the 10 year period?

Create a function that accepts the principal p, the term in years t, the interest rate r, and the number of compounding periods per year n. The function returns the value at the end of term rounded to the nearest cent.

For the example above:

compound\_interest(10000, 10, 0.06, 12) ➞ 18193.97

Note that the interest rate is given as a decimal and n=12 because with monthly compounding there are 12 periods per year. Compounding can also be done annually, quarterly, weekly, or daily.

### Examples

compound\_interest(100, 1, 0.05, 1) ➞ 105.0

compound\_interest(3500, 15, 0.1, 4) ➞ 15399.26

compound\_interest(100000, 20, 0.15, 365) ➞ 2007316.26

|  |
| --- |
| def compound\_interest(P, r, n, t):      return P\*(1 + (r/n))\*\*(n\*t)  # sample  print(compound\_interest(100, 1, 0.05, 1)) |

Question5

Write a function that takes a list of elements and returns only the integers.

### Examples

return\_only\_integer([9, 2, "space", "car", "lion", 16]) ➞ [9, 2, 16]

return\_only\_integer(["hello", 81, "basketball", 123, "fox"]) ➞ [81, 123]

return\_only\_integer([10, "121", 56, 20, "car", 3, "lion"]) ➞ [10, 56, 20, 3]

return\_only\_integer(["String", True, 3.3, 1]) ➞ [1]

|  |
| --- |
| l= [1, 2, 'a', 'b']  s= [i for i in l if type(i) == int]  print(s) |

Question1

Create a function that takes three parameters where:

* x is the start of the range (inclusive).
* y is the end of the range (inclusive).
* n is the divisor to be checked against.

Return an ordered list with numbers in the range that are divisible by the third parameter n. Return an empty list if there are no numbers that are divisible by n.

**Examples**

list\_operation(1, 10, 3) ➞ [3, 6, 9]

list\_operation(7, 9, 2) ➞ [8]

list\_operation(15, 20, 7) ➞ []

|  |
| --- |
| def list\_operation(x, y, n):      num\_list= [i for i in range(x, y+1) if i%n == 0]      return num\_list  # sample  print(list\_operation(1, 10, 3)) |

Question2

Create a function that takes in two lists and returns True if the second list follows the first list by **one** element, and False otherwise. In other words, determine if the second list is the first list shifted to the right by 1.

**Examples**

simon\_says([1, 2], [5, 1]) ➞ True

simon\_says([1, 2], [5, 5]) ➞ False

simon\_says([1, 2, 3, 4, 5], [0, 1, 2, 3, 4]) ➞ True

simon\_says([1, 2, 3, 4, 5], [5, 5, 1, 2, 3]) ➞ False

**Notes**

* Both input lists will be of the same length, and will have a minimum length of 2.
* The values of the 0-indexed element in the second list and the n-1th indexed element in the first list do not matter.

|  |
| --- |
|  |

Question3

A group of friends have decided to start a secret society. The name will be the first letter of each of their names, sorted in alphabetical order.

Create a function that takes in a list of names and returns the name of the secret society.

### Examples

society\_name(["Adam", "Sarah", "Malcolm"]) ➞ "AMS"

society\_name(["Harry", "Newt", "Luna", "Cho"]) ➞ "CHLN"

society\_name(["Phoebe", "Chandler", "Rachel", "Ross", "Monica", "Joey"])

|  |
| --- |
| def society\_name(names):      society\_name= ''      names.sort()      for i in names:          society\_name+= i[0]      return society\_name  # sample  print(society\_name(["Adam", "Sarah", "Malcolm"])) |

Question4

An isogram is a word that has no duplicate letters. Create a function that takes a string and returns either True or False depending on whether or not it's an "isogram".

**Examples**

is\_isogram("Algorism") ➞ True

is\_isogram("PasSword") ➞ False

# Not case sensitive.

is\_isogram("Consecutive") ➞ False

**Notes**

* Ignore letter case (should not be case sensitive).
* All test cases contain valid one word strings.

|  |
| --- |
| def is\_isogram(word):      word= word.lower()      if len(word) == len(set(word)):          return True      return False  # sample  print(is\_isogram("Algorism"))  print(is\_isogram("PasSword")) |

Question5

Create a function that takes a string and returns True or False, depending on whether the characters are in order or not.

### Examples

is\_in\_order("abc") ➞ True

is\_in\_order("edabit") ➞ False

is\_in\_order("123") ➞ True

is\_in\_order("xyzz") ➞ True

### Notes

You don't have to handle empty strings.

|  |
| --- |
| def is\_in\_order(string):      chars= [i for i in string]      chars.sort()      chars= ''.join(chars)      if string == chars:          return True      return False  # sample  print(is\_in\_order("abc"))  print(is\_in\_order("edabit")) |

Question 1

Create a function that takes a number as an argument and returns True or False depending on whether the number is symmetrical or not. A number is symmetrical when it is the same as its reverse.

**Examples**

is\_symmetrical(7227) ➞ True

is\_symmetrical(12567) ➞ False

is\_symmetrical(44444444) ➞ True

is\_symmetrical(9939) ➞ False

is\_symmetrical(1112111) ➞ True

|  |
| --- |
| def is\_symmetrical(n):      n= str(n)      if n == (n)[::-1]:          return True      return False  # sample  print(is\_symmetrical(7227))  print(is\_symmetrical(1256)) |

Question 2

Given a string of numbers separated by a comma and space, return the product of the numbers.

### Examples

multiply\_nums("2, 3") ➞ 6

multiply\_nums("1, 2, 3, 4") ➞ 24

multiply\_nums("54, 75, 453, 0") ➞ 0

multiply\_nums("10, -2") ➞ -20

|  |
| --- |
| def multiply\_nums(string):      num\_string= [int(i) for i in string if i.isdigit()]      r= 1      for i in num\_string:          r\*= int(i)      return r  # sample  print(multiply\_nums("1, 2, 3, 4")) |

Question 3

Create a function that squares every digit of a number.

### Examples

square\_digits(9119) ➞ 811181

square\_digits(2483) ➞ 416649

square\_digits(3212) ➞ 9414

### Notes

The function receives an integer and must return an integer.

|  |
| --- |
| def square\_digits(n):      r= ''      n= str(n)      for i in n:          r+= str(int(i)\*\*2)      return r  # sample  print(square\_digits(2483)) |

Question 4

Create a function that sorts a list and removes all duplicate items from it.

### Examples

setify([1, 3, 3, 5, 5]) ➞ [1, 3, 5]

setify([4, 4, 4, 4]) ➞ [4]

setify([5, 7, 8, 9, 10, 15]) ➞ [5, 7, 8, 9, 10, 15]

setify([3, 3, 3, 2, 1]) ➞ [1, 2, 3]

|  |
| --- |
| def setify(array):      array= list(set(array))      array.sort()      return array  # sample  print(setify([1, 3, 3, 5, 5])) |

Question 5

Create a function that returns the mean of all digits.

### Examples

mean(42) ➞ 3

mean(12345) ➞ 3

mean(666) ➞ 6

### Notes

* The mean of all digits is the sum of digits / how many digits there are (e.g. mean of digits in 512 is (5+1+2)/3(number of digits) = 8/3=2).
* The mean will always be an integer.

|  |
| --- |
| def mean(num):      num\_string= [int(i) for i in str(num)]      return int(sum(num\_string)/len(num\_string))  # sample  print(mean(42)) |

Question1

Create a function that takes an integer and returns a list from 1 to the given number, where:

1. If the number **can be divided** evenly by 4, amplify it by 10 (i.e. return 10 times the number).
2. If the number **cannot be divided** evenly by 4, simply return the number.

**Examples**

amplify(4) ➞ [1, 2, 3, 40]

amplify(3) ➞ [1, 2, 3]

amplify(25) ➞ [1, 2, 3, 40, 5, 6, 7, 80, 9, 10, 11, 120, 13, 14, 15, 160, 17, 18, 19, 200, 21, 22, 23, 240, 25]

**Notes**

* The given integer will always be equal to or greater than 1.
* Include the number (see example above).
* To perform this problem with its intended purpose, try doing it with list comprehensions. If that's too difficult, just solve the challenge any way you can.

|  |
| --- |
| def amplify(n):      return [i\*10 if i%4==0 else i for i in range(1, n+1)]  # sample  print(amplify(4))  print(amplify(25)) |

Question2

Create a function that takes a list of numbers and return the number that's unique.

### Examples

unique([3, 3, 3, 7, 3, 3]) ➞ 7

unique([0, 0, 0.77, 0, 0]) ➞ 0.77

unique([0, 1, 1, 1, 1, 1, 1, 1]) ➞ 0

### Notes

Test cases will always have exactly one unique number while all others are the same.

|  |
| --- |
| def unique(array):      dic= {}      for i in array:          if i in dic:              dic[i]+= 1          else:              dic[i] = 1      result= sorted(dic.items(), key= lambda x: x[1])      return result[0][0]  # sample  print(unique([3, 3, 3, 7, 3, 3])) |

Question3

Your task is to create a Circle constructor that creates a circle with a radius provided by an argument. The circles constructed must have two getters getArea() (PIr^2) and *getPerimeter()* (2PI\*r) which give both respective areas and perimeter (circumference).

For help with this class, I have provided you with a Rectangle constructor which you can use as a base example.

### Examples

circy = Circle(11)

circy.getArea()

# Should return 380.132711084365

circy = Circle(4.44)

circy.getPerimeter()

# Should return 27.897342763877365

### Notes

Round results up to the nearest integer.

|  |
| --- |
| import math  class Circle:      def \_\_init\_\_(self, r):          self.radius= r      @property      def getArea(self):          return math.pi\*self.radius\*\*2      @property      def getPerimeter(self):          return 2\*math.pi\*self.radius  # sample  circle= Circle(5)  print(circle.getArea)  print(circle.getPerimeter) |

Question4

Create a function that takes a list of strings and return a list, sorted from shortest to longest.

### Examples

sort\_by\_length(["Google", "Apple", "Microsoft"])

➞ ["Apple", "Google", "Microsoft"]

sort\_by\_length(["Leonardo", "Michelangelo", "Raphael", "Donatello"])

➞ ["Raphael", "Leonardo", "Donatello", "Michelangelo"]

sort\_by\_length(["Turing", "Einstein", "Jung"])

➞ ["Jung", "Turing", "Einstein"]

### Notes

All test cases contain lists with strings of different lengths, so you won't have to deal with multiple strings of the same length.

|  |
| --- |
| def sort\_by\_length(array):      len\_dict= {x:len(x) for x in array}      sorted\_len\_dict= sorted(len\_dict.items(), key= lambda x: x[1])      result= [i[0] for i in sorted\_len\_dict]      return result  # sample  print(sort\_by\_length(["Google", "Apple", "Microsoft"])) |

Question5

Create a function that validates whether three given integers form a **Pythagorean triplet**. The sum of the squares of the two smallest integers must equal the square of the largest number to be validated.

### Examples

is\_triplet(3, 4, 5) ➞ True

# 3² + 4² = 25

# 5² = 25

is\_triplet(13, 5, 12) ➞ True

# 5² + 12² = 169

# 13² = 169

is\_triplet(1, 2, 3) ➞ False

# 1² + 2² = 5

# 3² = 9

### Notes

Numbers may not be given in a sorted order.

|  |
| --- |
| def is\_triplet(x, y, z):      if x\*\*2 + y\*\*2 == z\*\*2:          return True      return False  # sample  print(is\_triplet(3, 4, 5)) |

Question1

Create a function that takes three integer arguments (a, b, c) and returns the amount of integers which are of equal value.

**Examples**

equal(3, 4, 3) ➞ 2

equal(1, 1, 1) ➞ 3

equal(3, 4, 1) ➞ 0

**Notes**

Your function must return 0, 2 or 3.

|  |
| --- |
| def equal(a, b, c):      if a == b == c:          return 3      elif a == b != c or a != b == c or a == c != b:          return 2      else:          return 0  # sample  print(equal(3, 4, 3)) |

Question2

Write a function that converts a **dictionary** into a **list** of keys-values **tuples**.

### Examples

dict\_to\_list({

"D": 1,

"B": 2,

"C": 3

}) ➞ [("B", 2), ("C", 3), ("D", 1)]

dict\_to\_list({

"likes": 2,

"dislikes": 3,

"followers": 10

}) ➞ [("dislikes", 3), ("followers", 10), ("likes", 2)]

### Notes

Return the elements in the list in alphabetical order.

|  |
| --- |
| d= {"likes": 2, "dislikes": 3, "followers": 10}  k= (list(d.keys()))  v= (list(d.values()))  print(list(zip(k, v))) |

Question3

Write a function that creates a dictionary with each **(key, value)** pair being the **(lower case, upper case)** versions of a letter, respectively.

### Examples

mapping(["p", "s"]) ➞ { "p": "P", "s": "S" }

mapping(["a", "b", "c"]) ➞ { "a": "A", "b": "B", "c": "C" }

mapping(["a", "v", "y", "z"]) ➞ { "a": "A", "v": "V", "y": "Y", "z": "Z" }

### Notes

All of the letters in the input list will always be lowercase.

|  |
| --- |
| def mapping(iterables):      return {x: x.upper() for x in iterables}  # sample  print(mapping(["p", "s"])) |

Question4

Write a function, that replaces all vowels in a string with a specified vowel.

### Examples

vow\_replace("apples and bananas", "u") ➞ "upplus und bununus"

vow\_replace("cheese casserole", "o") ➞ "chooso cossorolo"

vow\_replace("stuffed jalapeno poppers", "e") ➞ "steffed jelepene peppers"

### Notes

All words will be lowercase. Y is not considered a vowel.

|  |
| --- |
| def replace\_vowels(string, character):      vowels= 'aeiou'      output=''      for i in string:          if i in vowels:              output+= character          else:              output+= i      return output  # sample  print(replace\_vowels("the aardvark", "#")) |

Question5

Create a function that takes a string as input and capitalizes a letter if its ASCII code is even and returns its lower case version if its ASCII code is odd.

### Examples

ascii\_capitalize("to be or not to be!") ➞ "To Be oR NoT To Be!"

ascii\_capitalize("THE LITTLE MERMAID") ➞ "THe LiTTLe meRmaiD"

ascii\_capitalize("Oh what a beautiful morning.") ➞ "oH wHaT a BeauTiFuL moRNiNg."

|  |
| --- |
| def ascii\_capitalize(string):      result=''      array= [i for i in string]      for i in array:          if i.isalpha():              if ord(i)%2 == 0:                  result+= i.upper()              else:                  result+= i.lower()          else:              result+= i      return result  # sample  print(ascii\_capitalize("to be or not to be!")) |