Problem-1: Write a Python function that takes a list and returns a new list with unique elements of the first list.

Exercise 1:

Input:

```
[1,2,3,3,3,3,4,5]
```

Output:

```
[1, 2, 3, 4, 5]
```

```
# Write code here
def return_unique(L):
  res = []

  for i in L:
    if i not in res:
      res.append(i)

  return res

L = [1,2,3,3,3,3,4,5]
return_unique(L)
```

```
[1, 2, 3, 4, 5]
```

Problem-2: Write a Python function that accepts a hyphen-separated sequence of words as parameter and returns the words in a hyphen-separated sequence after sorting them alphabetically.

Example 1:

Input:

```
green-red-yellow-black-white
```

Output:

```
black-green-red-white-yellow
```

```
# Write code here

def sort_sequence(seq):
  temp = []

  for i in sorted(seq.split('-')):
    temp.append(i)

  return "-".join(temp)

s = 'green-red-yellow-black-white'
sort_sequence(s)
```

```
'black-green-red-white-yellow'
```

Problem 3: Write a Python function that accepts a string and calculate the number of upper case letters and lower case letters.

```
Sample String : 'CampusX is an Online Mentorship Program fOr EnginEering studentS.'
Expected Output :
```

```
No. of Upper case characters :  9
No. of Lower case Characters :  47
```

```
# Write code here

def lower_upper(s):
  lower_count = 0
  upper_count = 0

  for i in s:
    if i.islower():
      lower_count += 1
    elif i.isupper():
      upper_count += 1
    else:
      pass

  return lower_count,upper_count

s = 'CampusX is an Online Mentorship Program fOr EnginEering studentS.'
x,y = lower_upper(s)
print('No. of Lower case characters:', x)
print('No. of Upper case Characters:', y)
```

```
No. of Lower case characters: 47
No. of Upper case Characters: 9
```

Problem 4: Write a Python program to print the even numbers from a given list.

```
Sample List : [1, 2, 3, 4, 5, 6, 7, 8, 9]
Expected Result : [2, 4, 6, 8]
```

```
# Write code here

def is_even(L):
  res = []

  for i in L:
    if i % 2 == 0:
      res.append(i)

  return res

is_even([1,2,3,4,5,6,7])
```

```
[2, 4, 6]
```

Problem 5: Write a Python function to check whether a number is perfect or not.

A Perfect number is a number that is half the sum of all of its positive divisors (including itself).

Example :

```
The first perfect number is 6, because 1, 2, and 3 are its proper positive divisors, and 1 + 2 + 3 = 6.
Equivalently, the number 6 is equal to half the sum of all its positive divisors: ( 1 + 2 + 3 + 6 ) / 2 = 6.

The next perfect number is 28 = 1 + 2 + 4 + 7 + 14. This is followed by the perfect numbers 496 and 8128.
```

```
# Write code here
def perfect_num(num):

  sum = 0

  for i in range(1,num):
    if num % i == 0:
      sum += i

  return sum == num

perfect_num(29)
```

```
False
```

Problem-6: Write a Python function to concatenate any no of dictionaries to create a new one.

```
Sample Dictionary :
dic1={1:10, 2:20}
dic2={3:30, 4:40}
dic3={5:50,6:60}
Expected Result : {1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}
```

```
# Write code here

def merge_dict(*kwargs):
  d = {}

  for i in kwargs:
    d.update(i)

  return d


dic1={1:10, 2:20}
dic2={3:30, 4:40}
dic3={5:50,6:60}

merge_dict(dic1,dic2,dic3)
```

```
{1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}
```

Problem-7 Write a python function that accepts a string as input and returns the word with most occurence.

```
Input:
hello how are you i am fine thank you


Output
you -> 2
```

```
# Write code here
def most_used(s):

  d = {}
  for i in s.split():
    if i in d:
      d[i] = d[i] + 1
    else:
      d[i] = 1

  max_val = max(d.values())

  for i in d:
    if d[i] == max_val:
      print(i,'->',d[i])
      break

most_used('hello hello hello you i am fine thank you')
```

```
hello -> 3
```

`Problem-8` Write a python function that receives a list of integers and prints out a histogram of bin size 10

```
Input:
[13,42,15,37,22,39,41,50]


Output:
{11-20:2,21-30:1,31-40:2,41-50:3}
```

```
# Write code here
import math

def histogram(L):

  min_bin = math.floor(min(L)/10)*10
  max_bin = math.ceil(max(L)/10)*10

  d={}

  for i in range(min_bin,max_bin,10):
    count = 0
    for j in L:
      if i+1<=j<=i+10:
        count+=1

    d[str(i+1) + '-' + str(i+10)] = count

  return d

histogram([13,42,15,37,22,39,41,50])
```

```
{'11-20': 2, '21-30': 1, '31-40': 2, '41-50': 3}
```

`Problem-9` Write a python function that accepts a list of 2D co-ordinates and a query point, and then finds the the co-ordinate which is closest in terms of distance from the query point.

```
List of Coordinates
[(1,1),(2,2),(3,3),(4,4)]
Query Point
(0,0)
```

```
Output
Nearest to (0,0) is (1,1)
```

```
# Write code here

def shortest_dist(points,query):
  temp = []
  for i in points:
    distance = ((i[0] - query[0])**2 + (i[1] - query[1])**2)**0.5
    temp.append(distance)

  return points[sorted(list(enumerate(temp)),key=lambda x:x[1])[0][0]]


points = [(1,4),(2,-2),(13,3),(14,4)]
query = (0,0)

shortest_dist(points,query)
```

```
(2, -2)
```

Problem-10 :Write a python program that receives a list of strings and performs bag of word operation on those strings

https://en.wikipedia.org/wiki/Bag-of-words_model

```
# Write code here

def bow(L):

  vocab = set()

  for i in L:
    vocab.update(i.split())

  result = []

  for i in L:
    result.append([])
    for j in vocab:
      result[-1].append(i.count(j))

  print(vocab)
  return result


L = [
    'cat mat rat cat',
     'sat bat fat cat rat',
     'pat cat mat rat'
]

bow(L)
```

```
{'bat', 'sat', 'mat', 'cat', 'pat', 'rat', 'fat'}
[[0, 0, 1, 2, 0, 1, 0], [1, 1, 0, 1, 0, 1, 1], [0, 0, 1, 1, 1, 1, 0]]
```

Problem 11: Write a Python program to add three given lists using Python map and lambda.

```
# Write code here
L1 = [1,2,3]
L2 = [4,5,6]
L3 = [7,8,9]

# [12,15,18]

list(map(lambda x,y,z:x+y+z,L1,L2,L3))
```

```
[12, 15, 18]
```

`Problem-12`: Write a Python program to create a list containing the power of said number in bases raised to the corresponding number in the index using Python map.

```
Input:

 list1 = [1,2,3,4,5,6]

Output:

 [1,2,9,64,625,-]
```

```
# Write code here
list1 = [1,2,3,4,5,6]
list(map(lambda x,y:x**y,list1,range(len(list1))))
```

```
[1, 2, 9, 64, 625, 7776]
```

`Problem-13` Using filter() and list() functions and .lower() method filter all the vowels in a given string.

```
# Write code here
str1="FIFA world cup in 2022 will take place in Qatar"

list(filter(lambda x:True if x.lower() in 'aeiou' else False,str1))
```

```
['I', 'A', 'o', 'u', 'i', 'i', 'a', 'e', 'a', 'e', 'i', 'a', 'a']
```

`Problem-14` : Use reduce to convert a 2D list to 1D

```
# Write code here
ini_list = [[1, 2, 3],
            [3, 6, 7],
            [7, 5, 4]]

import functools
functools.reduce(lambda x,y:x+y,ini_list)
```

```
[1, 2, 3, 3, 6, 7, 7, 5, 4]
```

`Problem 15` - A dictionary contains following information about 5 employees:

- First name
- Last name
- Age
- Grade(Skilled,Semi-skilled,Highly skilled)
  Write a program using map/filter/reduce to a list of employees(first name + last name) who are highly skilled

```
# Write code here
employees = [
    {
        'fname':'Nitish',
        'lname':'Singh',
        'age' : 33,
        'grade':'skilled'
    },
    {
        'fname':'Ankit',
        'lname':'Verma',
        'age' : 34,
        'grade':'semi-skilled'
    },
    {
        'fname':'Neha',
```