LIST Questions

1. Given a list of numbers, write a Python program to find the sum of all the elements in the list.?

Input: arr = [2,4,5,10], i = 1, j = 3

Output: 19

Input: arr = [4,10,5,3,3], i = 3, j = 3

Output: 3

Solution:

```
n=int(input("Enter Length of array:"))
a=int(input("Enter the first element:"))
b=int(input("Enter the Last Element:"))

#Creating an Empty List
l=[]
#Adding elements into list
for i in range(0,n):
    e=int(input("Enter the element:"))
    l.append(e)
#Craeting a loop for summation in the given interval of (a,b)
sum=0
for j in range(a,b+1):
    sum=l[j]+sum

print(sum)
```

2. Given an array of integers arr[] of size N and an integer, the task is to rotate the array elements to the left by d positions.?

Input: arr[] = {1, 2, 3, 4, 5, 6, 7}, d = 2 Output: 3 4 5 6 7 1 2

Input: arr[] = {3, 4, 5, 6, 7, 1, 2}, d=2 Output: 5 6 7 1 2 3 4

Solution:

```
n=int(input("Enter length of list:"))
l=[]
for i in range(0,n):
    e=int(input("Enter an element:"))
    l.append(e)
f=[]
d=int(input('Enter the Rotating Element:'))
f=l[d:]+l[0:d]
print(f)
```

3. Second most repeated word in a sequence in Python? Given a sequence of strings, the task is to find out the second most repeated (or frequent) string in the given sequence.

Input : ["aaa", "bbb", "ccc", "bbb", "aaa", "aaa"]

Output : bbb

Solution:

```
l=[]
n=int(input('Enter the list size:'))

for i in range(n):
    l.append(input('Enter the elements:'))

d={}
```

```
for a in l:
    if a in d:
        d[a]+=1
    else:
        d[a]=1

f=sorted(d.items(),key=lambda x :x[1],reverse=True)

final=list(f[1])

print(final[0])
```

4. Difference between two lists ?

Input: list1 = [10, 15, 20, 25, 30, 35, 40]
list2 = [25, 40, 35]

Output: [10, 20, 30, 15]

Solution:

```
l1=[]
l2=[]
r=[]

n1=int(input("Enter first list lenght:"))
n2=int(input('Enter 2nd list lenght:'))
print("Please enter 1st list elements")
for i in range(0,n1):
    e=int(input("Enter the element:"))
    l1.append(e)
```

```
print("Please enter 2nd list elements")
for j in range(0,n2):
    E=int(input("Enter the elements:"))
    l2.append(E)

if n1>n2:
    for i in range(0, n1):
        for j in range(0, n2):
            if (l1[i] not in l2):
                r.append(l1[i])

else:
    for i in range(0, len(l2)):
        for j in range(0, len(l1)):
            if (l2[i] not in l1):
                r.append(l2[i])

print(list(set(r)))
```

5. Print all positive numbers from given list using for loop Iterate each element in the list using for loop and check if number is greater than or equal to 0. If the condition satisfies, then only print the number.?
Python program to print positive Numbers in a List

Input: list1 = [12, -7, 5, 64, -14] Output: 12, 5, 64

Input: list2 = [12, 14, -95, 3] Output: [12, 14, 3]

Solution:

```
n=int(input("Enter length of list:"))
l=[]
r=[]
for i in range(0,n):
    e=int(input("Enter the elements:"))
    l.append(e)
for i in range(0,n):
    if (l[i]>0):
        r.append(l[i])

print(r)
```

6. Write a Python program to flatten a given nested list structure.?

Original list: [0, 10, [20, 30], 40, 50, [60, 70, 80], [90, 100, 110, 120]]

Flatten list: [0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120]

Solution:

```
n=int(input("Enter the Main list length:"))
l=[]
for i in range(0,n):
    n1=int(input("Enter the sub-list length:"))
    a=[]
    for j in range(0,n1):
        e=int(input("Enter the elements:"))
        a.append(e)
    l.append(a)
```

```
r=[]
for k in range(0,n):
    for t in l[k]:
        r.append(t)

print(r)
```

7. Given an array and a value, find if there is a triplet in array whose sum is equal to the given value. If there is such a triplet present in array, then print the triplet and return true. Else return false.? Input: array = [12, 3, 4, 1, 6, 9], sum = 24; Output: 12, 3, 9 Explanation: There is a triplet (12, 3 and 9) present in the array whose sum is 24.

Input: array = [1, 2, 3, 4, 5], sum = 9

Output: 5, 3, 1

Explanation: There is a triplet (5, 3 and 1) present in the array whose sum is 9.

Solution:

```
n=int(input("Enter the length of list:"))
l=[]
r=[]
for i in range(0,n):
    e=int(input("Enter the elements:"))
    l.append(e)
sum=int(input("Enter the sum value:"))

for i in range(0,n):
    for j in range(i+1,n):
        for k in range(j+1,n):
            if sum==l[i]+l[j]+l[k]:
                print("There is a triplet ",'(',l[i],',',l[j],',',l[k],')','whose sum is',sum)
```

Strings

1. Missing characters to make a string Pangram.? Basic Questions 3 Pangram is a sentence containing every letter in the English alphabet. Given a string, find all characters that are missing from the string, i.e., the characters that can make the string a Pangram. We need to print output in alphabetic order. Input : welcome to geeksforgeeks

Output : abdhijnpquvxyz

Input : The quick brown fox jumps

Output : adglvyz

Solution:

```
si=input("Enter the input string:")
validate='abcdefghijlmnopqrstuvwxyz'
e=""
si=si.lower()
for i in range(len(validate)):
   if  validate[i] not in si:
      e=e+validate[i]

print(e)
```

2. Find total number of non-empty substrings of a string with N characters.?

Input : str = “abc” Output : 6 Every substring of the given string : “a”, “b”, “c”, “ab”, “bc”, “abc” Input : str = “abcd” Output : 10 Every substring of the given string : “a”, “b”, “c”, “d”, “ab”, “bc”, “cd”, “abc”, “bcd” and “abcd”

Solution:

```
s=input("Enter the string:")
sum=0
n=len(s)
```

```
for i in range(n):
    if i==1:
        sum=n+(n-1)
    elif i>1:
        sum=sum+(n-i)

for a in range(len(s)):
    for b in range(a,len(s)+1):
        print(s[a:b],end=' ')

print(sum)
```

4. Write a Python program that accepts a comma separated sequence of words as input and prints the unique words in sorted form (alphanumerically). ?

Sample Words : red, white, black, red, green, black

Expected Result : black, green, red, white,red

Solution:

```
s=input("Enter the strings:")

s=s.split(',')

l=sorted(tuple(s))

print(','.join(l))
```

5. Write a Python program to count the number of characters (character frequency) in a string. ?

Sample String : google.com' Expected Result : {'g': 2, 'o': 3, 'l': 1, 'e': 1, '.': 1, 'c': 1, 'm': 1}

Solution:

```
from collections import Counter
s=input('Enter the input string:')
'''
#1st Method
r=Counter(s)
print(r)
'''
#2nd Method

l={}
for i in s:
    if i in l:
        l[i]+=1
    else:
        l[i]=1

print(l)
```

6. find the frequency of minimum occurring character in a python string ?

The original string is : iNeuronNet.com The minimum of all characters in GeeksforGeeks is : I

Solution:

```
s=input("Enter the string:")
split_string=list(s)


d={}
for i in split_string:
    if i in d:
        d[i]+=1
    else:
        d[i]=1


final=sorted(d.items(),key=lambda x:x[1])


f=list(final[0])
print('The minimum of all Characters is',f[0])
```

7 Write a program extract all the string characters which have odd number of occurrences.

The original string is : geekforgeeks is best for geeks The Odd Frequency Characters are : ['k', 'i', 't', 'g', 'e', 'b']

Solution:

```
s=input('Enter the input string:')


l={}
for i in s:
    if i in l:
        l[i]+=1
```

```
    else:
        l[i]=1
r=[]
for a,b in l.items():
    if b%2==1:
        r.append(a)

print('The Odd frequency characters are :',r)
```

3. Given a string containing lowercase and uppercase letters. Sort it in such a manner that the uppercase and lowercase letters come in an alternate manner but in a sorted way.?

Input : bAwutndekWEdkd

Output :AbEdWddekkntuw

Explanation: Here we can see that letter ‘A’, ’E’, ’W’ are sorted as well as letters “b, d, d, d, e, k, k, n, t, u, w” are sorted but both appears alternately in the string as far as possible.

Input :abbfDDhGFBvdFDGBNDasZVDFjkb

Output :BaBaDbDbDbDdDfFhFjFkGsGvNVZ

Solution:

```
inp=input()
upper = sorted([x for x in inp if x > "A" and x ≤ "Z"] , reverse=True)
lower =sorted([x for x in inp if x > "a" and x < "z"], reverse=True)
result = [ ]
while (upper and lower):

u = upper.pop()
1 = lower.pop()
result.extend([u, l])
```

```
if upper:

result.extend(upper[:: -1])

if lower:

result.extend(lower [:: -1])



out = "".join(result)

print(out)
```

Dictionary

1. Given an input string and a pattern, check if characters in the input string follows the same order as determined by characters present in the pattern. Assume there won't be any duplicate characters in the pattern.?

Input: string = "engineers rock" pattern = "er";

Output: true Explanation: All 'e' in the input string are before all 'r'. Input: string = "engineers rock" pattern = "gsr"; Output: false Explanation: There are one 'r' before 's' in the input string.

Solution:

```
s=input('Enter the input string:')

p=input('enter the pattern:')

l=[]

for i in p:
```

```
 l.append(i)


for j in range(0,len(l)-1):
    if len(p) > len(s):
        print('False')
        break
    x=l[j]
    y=l[j+1]


    first=s.index(x)
    last=s.index(y)


    if first==-1 or last==-1 or first>last:
        print('False')
        break
else:
    print('True')
```

2. Given a list and dictionary, map each element of list with each item of dictionary, forming nested dictionary as value.?

Input : test_dict = {‘Gfg’ : 4, ‘best’ : 9}, test_list = [8, 2]

Output : {8: {‘Gfg’: 4}, 2: {‘best’: 9}}

Explanation : Index-wise key-value pairing from list [8] to dict {‘Gfg’ : 4} and so on

Solution:

```
l=[]
d={}
n=int(input('Enter the list length:'))
```

```
for i in range(n):
    e=int(input('Enter the list element:'))
    l.append(e)

for j in range(n):
    k=input('Enter the key')
    d[k]=int(input('Enter the value:'))



o={}
for k,v in d.items():
    for i in l:
        f={k:v}
        o[i]=f
print(o)
```

3. Sort Dictionary key and values List ?

Input : test_dict = {‘c’: [3], ‘b’: [12, 10], ‘a’: [19, 4]}

Output : {‘a’: [4, 19], ‘b’: [10, 12], ‘c’: [3]}

Input : test_dict = {‘c’: [10, 34, 3]}

Output : {‘c’: [3, 10, 34]}

Solution:

```
l={}

n=int(input('Enter the number of keys:'))

for i in range(n):
    key=input('Enter the key:')
    temp=[]
```

```
    n1=int(input('Enter the list size:'))
    for j in range(n1):
        temp.append(int(input('Enter the elements into sub list:')))
    l[key]=temp

print(l)

t=dict(sorted(l.items()))
#Sorting the values
for k,v in t.items():
    if len(v)==1:
        t[k]=v
    else:
        for i in range(len(v)):
            for j in range(i+1,len(v)):
                temp=0
                if v[i]>v[j]:
                    temp=v[i]
                    v[i]=v[j]
                    v[j]=temp
                else:
                    pass
    t[k]=v

print(t)
```

4. Remove all duplicates words from a given sentence ?

Input : Python is great and Java is also great

Output : is also Java Python and great

Solution:

```
l=input('Enter the string:')
temp=l.split(' ')

d={}

for i in temp:
    if i in d:
        d[i]+=1

    else:
        d[i]=1

for j in d.keys():
    print(j,end=' ')
```

5. Inversion in nested dictionary?

Input : test_dict = {“a” : {“b” : {}}, “d” : {“e” : {}}, “f” : {“g” : {}}

Output : {‘b’: {‘a’: {}}, ‘e’: {‘d’: {}}, ‘g’: {‘f’: {}}

Explanation : Nested dictionaries inverted as outer dictionary keys and viz-a-vis.

Solution:

```
l={}
n=int(input('Enter the length of main dictionary:'))

for i in range(n):
    k=input('Enter the outer-key:')

    sub_dict={}
    k1=input('Enter the sub-key:')
    sub_dict[k1]={}

    l[k]=sub_dict



output = { a:{k:b} for k,v in l.items() for a,b in v.items() }
print(l)

print(output)
```

6. Given an array of n string containing lowercase letters. Find the size of largest subset of string which are anagram of each others. An anagram of a string is another string that contains same characters, only the order of characters can be different.

For example, “abcd” and “dabc” are anagram of each other.?

Input: ant magenta magnate tan gnamate

Output: 3 Explanation Anagram strings(1) - ant, tan Anagram

strings(2) - magenta, magnate, gnamate

Thus, only second subset have largest size i.e., 3 Input: cars bikes arcs steer Output: 2

Solution:

```
s=[]
n=int(input('enter the list size:'))
for a in range(n):
    s.append(input('Enter the elements:'))
s_list=[]

for i in range(len(s)):
    temp=list(s[i])
    temp.sort()
    s1=''
    for j in temp:
        s1+=j
    s_list.append(s1)
d={}
for k in s_list:
    if k in d:
        d[k]+=1
    else:
        d[k]=1

print(d)
```

```
t=sorted(d.items(),key=lambda x:x[1],reverse=True)
print(t[0][1])
```

Sets:

1. Given two lists a, b. Check if two lists have at least one element common in them.

Input : a = [1, 2, 3, 4, 5] b = [5, 6, 7, 8, 9]

Output : True

Input : a=[1, 2, 3, 4, 5] b=[6, 7, 8, 9]

Output : False

Solution:

```
l1=[]
l2=[]
n1=int(input("Enter the 1st List length:"))
n2=int(input("Enter the 2nd list length:"))
for a in range(0,n1):
    e=int(input("Enter the 1st list elements:"))
    l1.append(e)
for b in range(0,n2):
    q=int(input("Enter the 2nd list elements:"))
    l2.append(q)
for i in range(0,len(l1)):
    if l1[i] in l2:
        print('True')
        break
else:
    print("False")
```

2. Return a new set of identical items from two sets

set1 = {10, 20, 30, 40, 50} set2 = {30, 40, 50, 60, 70}

Expected output: {40, 50, 30}

Solution:

```
s1=set()
s2=set()
n1=int(input("Enter the length of 1st set:"))
n2=int(input("Enter the length of 2nd set:"))
r=set()

for i in range(0,n1):
    e=int(input('Enter the elements of 1st set:'))
    s1.add(e)

for j in range(0,n2):
    t=int(input('Enter the elements of 2nd set:'))
    s2.add(t)

for q in s1:
    if q in s2:
        r.add(q)

print(r)
```

3. Maximum and Minimum in a Set without use of inbuild max/min functions?

Input : set = ([8, 16, 24, 1, 25, 3, 10, 65, 55])

Output : max is 65

Input : set = ([4, 12, 10, 9, 4, 13])

Output : min is 4

Solution:

```
def maxi(w1):
    l1=list(w1)
    max=0
    for i in range(0,len(l1)):
        for j in range(i+1,len(l1)):
            if l1[i]>l1[j]:
                max=l1[i]
            else:
                max=l1[j]
    return max


def mini(w1):
    l2=list(w1)
    min=l2[0]
    for a in range(1,len(l2)):
        if min>l2[a]:
            min=l2[a]


    return min
```

```
s1=set()
n=int(input('Enter the length of set:'))
for i in range(0,n):
    e=int(input("Enter the elements:"))
    s1.add(e)


print(maxi(s1))
print(mini(s1))
```

4. Write a Python program to check if a set is a subset of another set

Input :- x: {'mango', 'apple'} y: {'mango', 'orange'} z: {'mango'}

output :- If x is subset of y False False

If y is subset of x False False

If y is subset of z False False

If z is subset of y True True

Solution:

```
s1=set()
s2=set()
s3=set()
n1=int(input("Enter the length of 1st set:"))
n2=int(input("Enter the length of 2nd set:"))
n3=int(input("Enter the length of 3rd set:"))
r=set()
for i in range(0,n1):
    e=int(input('Enter the elements of 1st set:'))
    s1.add(e)
```

```
for j in range(0,n2):
    t=int(input('Enter the elements of 2nd set:'))
    s2.add(t)

for k in range(0,n3):
    q=int(input('Enter the elements of 3rd set:'))
    s3.add(q)



print('If s1 is a subset of s2')
if (s1.issubset(s2)):
    print('True')
else:
    print('False')
print('If s1 is a subset of s3')
if (s1.issubset(s3)):
    print('True')
else:
    print('False')
print('If s2 is a subset of s1')
if (s2.issubset(s1)):
    print('True')
else:
    print('False')
print('If s2 is a subset of s3')
if (s2.issubset(s3)):
    print('True')
else:
```

```
    print('False')
print('If s3 is a subset of s1')
if (s3.issubset(s1)):
    print('True')
else:
    print('False')
print('If s3 is a subset of s2')
if (s3.issubset(s2)):
    print('True')
else:
    print('False')
```

5. Write a Python program to remove the intersection of a 2nd set from the 1st set

input:- Original sets: {1, 2, 3, 4, 5} {4, 5, 6, 7, 8}

Output:- sn1: {1, 2, 3} sn2: {4, 5, 6, 7, 8}

Solution:

```
s1=set()
s2=set()
n1=int(input("Enter the length of 1st set:"))
n2=int(input("Enter the length of 2nd set:"))
r=set()
for i in range(0,n1):
    e=int(input('Enter the elements of 1st set:'))
    s1.add(e)

for j in range(0,n2):
    t=int(input('Enter the elements of 2nd set:'))
    s2.add(t)
```

```
print(s1.difference(s2))
r=(s1.intersection(s2)).union(s2)
print(r)
```

6. What is the result of passing a dictionary to a set constructor?

Solution:

When a dictionary is passed to set constructor only the keys will be selected.

Tuple

1. Remove Tuples of Length K ?

Input : test_list = [(4, 5), (4,), (8, 6, 7), (1,), (3, 4, 6, 7)], K = 2

Output : [(4,), (8, 6, 7), (1,), (3, 4, 6, 7)] Explanation : (4, 5) of len = 2 is removed.

Solution:

```
l=[]
n=int(input("Enter the length of main list:"))

for i in range(n):
    n1=int(input("Enter the sub-tuple length:"))
    a=[]
    for j in range(n1):
        a.append(int(input("Enter the elements into tuple:")))
    l.append(tuple(a))

k=int(input('Enter the tuple size which is required for removal:'))
r=[]
for z in range(n):
    if(len(l[z])!=k):
        r.append(l[z])
```

```
print(r)
```

2. Removing duplicates from tuple ?

The original tuple is : (1, 3, 5, 2, 3, 5, 1, 1, 3) The tuple after removing duplicates : (1, 3, 5, 2)

Solution:

```
l = []
n = int(input("Enter the lenght of list:"))
for i in range(n):
    l.append(int(input("Enter the elements:")))

l = tuple(l)
print("Tuple:",l)
print("After removing duplicates",tuple(set(l)))
```

3. Flatten tuple of List to tuple ?

Input : test_tuple = ([5], [6], [3], [8])

Output : (5, 6, 3, 8)

Input : test_tuple = ([5, 7, 8])

Output : (5, 7, 8)

Solution:

```
l=[]
n=int(input("Enter the length of main list:"))

for i in range(n):
    n1=int(input("Enter the length of sub list:"))
    a=[]
```

```
    for j in range(n1):
        a.append(int(input('Enter the elements:')))
    l.append(a)

l=tuple(l)
r=[]
print("Un-flattened Tuple:",l)
for y in range(n):
    for z in l[y]:
        r.append(z)

print('Flattened Tuple',tuple(r))
```

4. Remove nested records from tuple?

The original tuple : (1, 5, 7, (4, 6), 10)

Elements after removal of nested records : (1, 5, 7, 10)

Solution:

```
l=[]
n=int(input("Enter the length of main list:"))
for i in range(n):
    n1=int(input("Enter the length of sub-list:"))
    if n1>1:
        a=[]
        for j in range(0,n1):
            a.append(int(input("Enter the elements:")))
        l.append(tuple(a))
    else:
        for o in range(0,n1):
            l.append(int(input("Enter the elements:")))
```

```
print('Before removal of nested elements:',tuple(l))
d=[]


dt=()
for p in range(len(l)):
    if type(l[p])!=type(dt):
        d.append(l[p])
print(tuple(d))
```

5. Convert Binary tuple to Integer? The original tuple is : (1, 1, 0, 1, 0, 0, 1) Decimal number is : 105

Solution:

```
l = []
n = int(input("Enter the lenght of tuple:"))
for i in range(n):
    l.append(int(input("Enter the elements:")))
print("The Binary elements:",tuple(l))
sum = 0

for i in range(n):
    sum=sum+((2**i)*l[i])

print("The decimal sum is",sum)
```

6. Sort Tuples by Total digits?

Input : test_list = [(3, 4, 6, 723), (1, 2), (134, 234, 34)] Output : [(1, 2), (3, 4, 6, 723), (134, 234, 34)]
Explanation : 2 < 6 < 8, sorted by increasing total digits.

Solution:

```
l=[]
```

```
n=int(input("Enter the length of main list:"))

for i in range(n):
    n1=int(input("Enter the sub-tuple length:"))
    a=[]
    for j in range(n1):
        a.append(int(input("Enter the elements into tuple:")))
    l.append(tuple(a))

for i in range(len(l)):
    for j in range(i + 1, len(l)):
        temp = 0
        if (len(l[i]) < len(l[j])):
            pass
        else:
            temp = l[i]
            l[i] = l[j]
            l[j] = temp

print(l)
```

MAP & Lambda Function

1. Write a Python program to add three given lists using Python map and lambda.?

Original list: [1, 2, 3] [4, 5, 6] [7, 8, 9] New list after adding above three lists: [12, 15, 18]

Solution:

```
l1=[]
l2=[]
l3=[]
```

```
n1=int(input('Enter the length of 1st list:'))
n2=int(input('Enter the length of 2nd list:'))
n3=int(input('Enter the length of 3rd list:'))

for i in range(n1):
    l1.append(int(input('Enter the elements of 1st list:')))

for j in range(n2):
    l2.append(int(input('Enter the elements of 2nd list:')))

for k in range(n3):
    l3.append(int(input('Enter the elements of 3rd list:')))

sum= lambda x,y,z:x+y+z

final=list(map(sum,l1,l2,l3))

print('The final sum of 3 lists:',final)
```

2. Write a Python program to convert a given list of tuples to a list of strings using map function?
Original list of tuples: [('red', 'pink'), ('white', 'black'), ('orange', 'green')] Convert the said list of tuples to a list of strings: ['red pink', 'white black', 'orange green']

Solution:

```
l=[]

n=int(input("Enter the length of list:"))

for i in range(n):
```

```
    n1=int(input("Enter the length of sub-tuple:"))
    a=[]
    for j in range(n1):

        a.append(input("Enter the elements:"))
    l.append(tuple(a))

p=list(map(' '.join,l))

print(p)
```

3. Write a Python program to add two given lists and find the difference between lists. Use map() function? Original lists: [6, 5, 3, 9] [0, 1, 7, 7] Result: [(6, 6), (6, 4), (10, -4), (16, 2)]

Solution:

```
l1=[]
l2=[]


n1=int(input('Enter the length of 1st list:'))
n2=int(input('Enter the length of 2nd list:'))


for i in range(n1):
    l1.append(int(input('Enter the elements of 1st list:')))

for j in range(n2):
    l2.append(int(input('Enter the elements of 2nd list:')))
```

```
Max_1=lambda x,y:x+y
Min_1=lambda x,y:x-y
r=[]

output_1=list(map(Max_1,l1,l2))
output_2=list(map(Min_1,l1,l2))
#print(output_1)
#print(output_2)

r=[]
for i in range(n1):
    e=output_1[i]
    f=output_2[i]
    g=(e,f)
    r.append(g)

print(r)
```

4. Write a Python program to create Fibonacci series upto n using Lambda?

Fibonacci series upto 2: [0, 1]

Fibonacci series upto 5: [0, 1, 1, 2, 3]

Solution:

```
l=[]
sum=lambda x,y:x+y
n=5
for i in range(n):
    if i==0:
        a=i
        l.append(i)
```

```
    elif i==1:
        b=i
        l.append(i)

    else:
        e=sum(a,b)
        l.append(e)
        a=b
        b=e
print(l)
```

5. Write a Python program to find intersection of two given arrays using Lambda ?

[1, 2, 3, 5, 7, 8, 9, 10] [1, 2, 4, 8, 9] Intersection of the said arrays: [1, 2, 8, 9]

Solution:

```
l1=[]
l2=[]


n1=int(input('Enter the length of 1st list:'))
n2=int(input('Enter the length of 2nd list:'))


for i in range(n1):
    l1.append(int(input('Enter the elements of 1st list:')))

for j in range(n2):
    l2.append(int(input('Enter the elements of 2nd list:')))

l1=set(l1)
```

```
l2=set(l2)


l=lambda x,y: x.intersection(y)


print(list(l(l1,l2)))
```

6. Write a Python program to find palindromes in a given list of strings using Lambda ?

Orginal list of strings: ['php', 'w3r', 'Python', 'abcd', 'Java', 'aaa'] List of palindromes: ['php', 'aaa']

Solution:

```
l=[]
n=int(input('Enter the length of 1st list:'))
for i in range(n):
    l.append(input('Enter the elements of into list:'))
r=[]


s=lambda w: w if w==w[::-1] else None


r=list(map(s,l))


final=[]
for u in range(len(r)):
    if r[u]!=None:
        final.append(r[u])
print(final)
```

7. Write a Python program to find the list with maximum and minimum length using lambda ? Original list: [[0], [1, 3], [5, 7], [9, 11], [13, 15, 17]] List with maximum length of lists: (3, [13, 15, 17]) List with minimum length of lists: (1, [0])

Solution:

```
l=[]
```

```
n=int(input("Enter the total size:"))

for i in range(n):
    n1=int(input("Enter the sub-list size:"))
    a=[]
    for j in range(n1):
        a.append(int(input("Enter the sub-list elements:")))
    l.append(a)

print("The user input is:",tuple(l))

max_lambda_element=lambda s:len(s)
#max_ele=max_lambda_element(l)
#print(len(max_ele),max_ele)

for a in range(n):
    for b in range(i+1,n):
        x=max_lambda_element(l[a])
        y=max_lambda_element(l[b])
        temp=0
        if x>y:
            pass
        elif x==y:
            pass
        else:
            temp=l[a]
            l[a]=l[b]
            l[b]=temp
```

```
print(tuple(l))
print('List with maximum length of lists','(',max_lambda_element(l[0]),',',l[0],')',sep='')
print('List with minimum length of lists','(',max_lambda_element(l[-1]),',',l[-1],')',sep='')

'''
l.sort()
print('(',len(l[-1]),',',l[-1],')')
'''
```

8. Write a Python program to triple all numbers of a given list of integers.?

Original list: (1, 2, 3, 4, 5, 6, 7) Triple of said list numbers: [3, 6, 9, 12, 15, 18, 21]

Solution:

List Comprehension

1. Use a nested list comprehension to find all of the numbers from 1–1000 that are divisible by any single digit besides 1 (2–9)?

Solution:

```
l=[i for i in range(1,1001) for j in range(2,10) if i%j==0 ]
print(list(set(l)))
```

2. Use list comprehension to construct a new list but add 6 to each item?

Solution:

```
l=[]

n=int(input('Enter the length of list:'))

for i in range(n):

    l.append(int(input('Enter the elements:')))


l1={}


l1=[(i,i**3) for i in l if i%2==1]

print(dict(l1))
```

3. Suppose we want to create an output dictionary which contains only the odd numbers that are present in the input list as keys and their cubes as values. Let's see how to do this using for loops and dictionary comprehension.? input = [1, 2, 3, 4, 5, 6, 7] outpur :- {1: 1, 3: 27, 5: 125, 7: 343}

Solution:

```
l=[]
```

```
n=int(input('Enter the length of list:'))
for i in range(n):
    l.append(int(input('Enter the elements:')))

l1=[i+6 for i in l]

print(l1)
```

4. Given two lists containing the names of states and their corresponding capitals, construct a dictionary which maps the states with their respective capitals. Let's see how to do this using for loops and dictionary comprehension.? state = ['Gujarat', 'Maharashtra', 'Rajasthan'] capital = ['Gandhinagar', 'Mumbai', 'Jaipur'] output:- {'Gujarat': 'Gandhinagar','Maharashtra': 'Mumbai', 'Rajasthan': 'Jaipur'}

Solution:

```
l1=[]
n1=int(input("Enter the size of 1st list:"))
l2=[]
n2=int(input('Enter the size of 2nd list:'))

for i in range(n1):
    l1.append(input('Enter the elements of 1st list:'))

for j in range(n2):
    l2.append(input('Enter the elements of 2nd list:'))

final={a:b for a,b in zip(l1,l2)}

print(final)
```

5. Transpose of Matrix using Comprehension? Input :- [[1, 2, 3, 4], [4, 5, 6, 8]] Output :- [[1, 4], [2, 5], [3, 6], [4, 8]]

Solution:

```
l=[]
```

```
n=int(input('Enter the length of list:'))


n1=int(input('Enter the length of sub-list:'))


for a in range(n):
    t=[]
    for b in range(n1):
        t.append(int(input('Enter the sub-list elements:')))
    l.append(t)
print(l)
z=[[l[j][i] for j in range(0,len(l))] for i in range(n)]
print(z)
```

6. We have a string ‘2459a09b’ and we want to extract all integer literals, and use int() to cast them into integers.? Input- 2459a09b' output :- [2, 4, 5, 9, 9]

Solution:

```
s=input('Enter the string:')
w='0123456789'
l=[]
l=[i for i in s if i in w]

for j in l:
    print(int(j),end='')
```

7. Finding the elements in a list in which elements are ended with the letter ‘b’ and the length of that element is greater than 2? input :- names = ['Ch','Dh','Eh','cb','Tb','Td','Chb','Tdb'] output :- ['Chb', 'Tdb']

Soultion:

```
d=[]
r=input('Enter the required character:')
n=int(input('Enter the length of list:'))
for i in range(n):
    d.append(input('Enter the elements:'))
#filtering the objects who's length is greater than 2
```

```
l=[i for i in d if len(i)>2]


#Checking the last element is matching with the respective element
for j in l:
    for k in reversed(range(len(j))):
        if j[k]==r and k==-1:
            print(j)
```

8. Reverse each String in a Tuple using list comprehension ? Input :- 'Hello', 'Analytics', 'Vidhya' output :- ['olleH', 'scitylanA', 'ayhdiV']

Solution:

```
l=[]
n=int(input('Enter the length of list:'))
for i in range(n):
    l.append(int(input('Enter the elements:')))


r=[i[::-1] for i in l]


print(r)
```

Exception Handling

1. Write Custom Exception to handle Zero division Error ?

Solution:

```
n=int(input('Enter any number:'))
```

```
try:
    print('Before Zerobyte error:')
    d=n/0
except ZeroDivisionError as er:
    print('Now we got Zerobyte error as any number cannot be divided by Zero.The following occured error is a',str(er),'error')
```

2. Catching specific exception in Python.

Solution:

```
n=int(input('Enter a number:'))

try:
    if(n%2==1):
        raise Exception('The following number is odd number')
    else:
        print(n)

except Exception as er:
    print('Failed execution as -->',str(er))
```

Object Oriented Programming

1. Define a property that must have the same value for every class instance (object)?

Output :- Color: White, Vehicle name: School Volvo, Speed: 180, Mileage: 12

Color: White, Vehicle name: Audi Q5, Speed: 240, Mileage: 18

Solution:

```
class Car:
    def __init__(self,Vehicle_name,Speed,Mileage):
        self.colour='White'
        self.name=Vehicle_name
        self.speed=Speed
        self.mileage=Mileage
        print(self.colour,Vehicle_name,Speed,Mileage)



car1=Car('BMW',10,78)
car2=Car('Benz',20,60)
```

2. Create a class with property decorator with setter and getter functions.?

Solution:

```
class Car:
    def __init__(self,name,Mileage):
        self.__name=name
        self.__mileage=Mileage
    @property
    def name(self):
        return self.__name
    @name.getter
    def name(self):
        return self.__name
    @property
    def Mileage(self):
        return self.__mileage

    @Mileage.getter
    def Mileage(self):
        return self.__mileage

    @Mileage.setter
```

```
    def Mileage(self,Mileage):
        self.__mileage=Mileage

c=Car('BMW',10)

print(c.name)



print(c.Mileage)
Fuel_Injector=10
print("After Fuel Injector")

c.Mileage+=Fuel_Injector
print(c.Mileage)
```

3. Create a class with Multi-level Inheritance ?

```
class Car:
    def __init__(self,company,mileage):
        self.company=company
        self.milegae=mileage

class Driver(Car):
    def __init__(self,company,mileage,engine,colour):
        self.engine=engine
        self.colour=colour
        Car.__init__(self,company,mileage)
        print('The Car details are as follows')

print('Company:',self.company,'Engine:',self.engine,'Mileage:',self.milegae,'Colour:',sel
f.colour)

test=Driver('BMW',24,'Petrol','White')
```

4. Create a class a overwrite a class Destructor ?

Solution:

```
class Destruct:

    def __init__(self):
        print('Instance is created')

    def __del__(self):
        print('Instance is deleted')

d=Destruct()

del d
```

5. write a decorator for a class?

6. Implement a stack ?

Solution:

```
class Stack:
    def __init__(self,n):
        self.li=[]
        self.max_elements=n

    def isempty(self):
        if (len(self.li)==0):
            return True
        else:
            return False
    def isfull(self):
        if (len(self.li)==self.max_elements):
            return True
        else:
            return False
    def push(self,a):
        return self.li.append(a)

    def pop(self):
        return self.li.pop()

    def display(self):
        return self.li



n=int(input('Enter the list size:'))
s=Stack(n)
while(True):
    choice=int(input('Enter your Choice of Operation: '
                '1)IsFull '
                '2)IsEmpty '
                '3)Push '
                '4)Pop '
                '5)Display '
                '6)Exit:'))
    if choice == 1:
        print('Checking for the stack is full or not')
        print(s.isfull())

    elif choice == 2:
        print('Checking for the stack is empty or not')
        print(s.isempty())
```

```
    elif choice == 3:
        print('Adding elements in stack')
        if s.isfull():
            print('The Stack is full')



        else:
            a = int(input('Enter the element:'))
            s.push(a)


    elif choice == 4:
        print('Removing elements in stack')
        if s.isempty():
            print('Stack is empty')
        else:
            print('Removed Element is:', s.pop())

    elif choice == 5:
        print('Displaying the Stack')

        print(s.display())

    elif choice == 6:
        break
```

7. Implement a Queue ?

Solution:

```
class Queue:
    def __init__(self,n):
        self.li=[]
        self.max_elements=n

    def isempty(self):
        if (len(self.li)==0):
            return True
        else:
            return False
    def isfull(self):
        if (len(self.li)==self.max_elements):
            return True
        else:
            return False
    def Enqueue(self,a):
```

```
        return self.li.append(a)

    def Dequeue(self):
        return self.li.pop(0)

    def display(self):
        return self.li



n=int(input('Enter the list size:'))
s=Queue(n)
while(True):
    choice=int(input('Enter your Choice of Operation: '
                 '1)IsFull '
                 '2)IsEmpty '
                 '3)Enqueue '
                 '4)Dequeue '
                 '5)Display '
                 '6)Exit:'))
    if choice == 1:
        print('Checking for the Queue is full or not')
        print(s.isfull())

    elif choice == 2:
        print('Checking for the Queue is empty or not')
        print(s.isempty())

    elif choice == 3:
        print('Adding elements in Queue')
        if s.isfull():
            print('The Queue is full')
        else:
            a = int(input('Enter the element:'))
            s.Enqueue(a)


    elif choice == 4:
        print('Removing elements in Queue')
        if s.isempty():
            print('Queue is empty')
        else:
            print('Removed Element is:', s.Dequeue())

    elif choice == 5:
        print('Displaying the Queue')

        print(s.display())

    elif choice == 6:
        break
```

8. Implement a Linked List ?

Solution:

```
class Node:
    head = None  # Class Variable

    def __init__(self, data):
        self.data = data  # Instance Variable's
        self.next = None


if __name__ == '__main__':
    L=[]
    n=int(input("Enter the Number of elements:"))
    for e in range(n):
        ele=int(input('Enter the element:'))
        L.append((ele))

    for i in L:
        temp = Node(i)
        print(temp.data, end=' ')
        if (Node.head == None):
            Node.head = temp
        else:
            temp.next = Node.head
            Node.head = temp
```

9. Explain What happens you create a object of a class (Inside python)?

Solution:

A particular object will be created according to the class which we have defined. At first, the constructor will be creating the initial objects mentioned in it.

Once the object for a particular class has been created it will have the access to create methods with respective to the Class.

10. Create a class whose object should be called like a function.?

class Product:

def __init__(self):

print("Instance Created")

Instance created ans = Product()

ans(10, 20) -> should return product

```
class Product:

    def __call__(self, n1, n2):
        print(n1*n2)

p=Product()

p(1,2)
```