In [1]:

```
#1.Write a Python program to sort a list of tuples using Lambda

l1 = [(7,8,9),(1,2,3),(4,5,6)]

result=lambda x : sorted(x)

result(l1)
```

Out[1]:

```
[(1, 2, 3), (4, 5, 6), (7, 8, 9)]
```

In [1]:

```
# 2. Write a Python program to sort a list of dictionaries using Lambda

list1 = [{"A":10,"B":20},{"A":44,"F":77},{"C":66,"A":14}]
print("original Lists of dict:",list1)

result = sorted(list1,key = lambda x: x['A'])

print("After sorting",result)
```

```
original Lists of dict: [{'A': 10, 'B': 20}, {'A': 44, 'F': 77}, {'C': 66,
'A': 14}]
After sorting [{'A': 10, 'B': 20}, {'C': 66, 'A': 14}, {'A': 44, 'F': 77}]
```

In [17]:

```
### 3. Write a Python program to find square and cube every number in a given list of integ


s = int(input("Enter the size of the list:"))
input_list = []
for i in range(s):
    item = int(input("Enter the list element:"))
    input_list.append(item)

square_list = list(map(lambda x: x**2,input_list))
cube_list = list(map(lambda x: x**3,input_list))

print("Square list is:",square_list)
print("Cube list is:",cube_list)
```

```
Enter the size of the list:4
Enter the list element:3
Enter the list element:6
Enter the list element:7
Enter the list element:9
Square list is: [9, 36, 49, 81]
Cube list is: [27, 216, 343, 729]
```

In [20]:

```
A=[2,5,7,4,9]

b = list(map(lambda x : x**2,A))
c = list(map(lambda x : x**3,A))
print("square of given list = ",b)
print("cube of given list = ",c)
```

```
square of given list =  [4, 25, 49, 16, 81]
cube of given list =  [8, 125, 343, 64, 729]
```

In [24]:

```
# 4. Write a Python program to find if a given string starts with a given character using L

str1="Data science and python"

A=lambda B : B.startswith(input("enter starting char = "))

A(str1)
```

```
enter starting char = Data
```

Out[24]:

```
True
```

In [22]:

```
starts_with = lambda x: True if x.startswith(input("Enter the char:-")) else False
print(starts_with(input("Enter the string:-")))
```

```
Enter the string:-python and data science
Enter the char:-python
True
```

In [25]:

```
#5. Write a Python program to check whether a given string is number or not using Lambda

s=input("Enter given input = ")

b=lambda x : x.isdigit()

print(b(s))
```

```
Enter given input = 1233
True
```

In [2]:

```
#6. Write a Python program to create Fibonacci series using Lambda

def fab(num):
    a = 0
    b = 1
    if num == 1:
        print(a)
    else:
        print(a)
        print(b)
        for i in range(2,num):
            c= a +b
            a = b
            b = c
            print(c)
fab(5)
```

```
0
1
1
2
3
```

In [27]:

```
#7. Write a Python program to find the intersection of two given arrays using Lambda
list1 = [2,3,4,5,6,11]
list2 = [7,8,9,2,4,3]

list3=list(filter(lambda x : x in list1,list2))

print("Intersection of the two list = ",list3)
```

```
Intersection of the two list =  [2, 4, 3]
```

In [33]:

```
#8. Write a Python program to rearrange positive and negative numbers in a given array usin

s = [-2,4,-5,3,5,-6,8,9,0]

b = lambda x : sorted(x)

print(b(s))
```

```
[-6, -5, -2, 0, 3, 4, 5, 8, 9]
```

In [34]:

```
#9. Write a Python program to count the even, odd numbers in a given array of integers usin

Arr1=[1,3,4,66,7,8,99,55,33]

even=list(filter(lambda x : x%2 == 0,Arr1))

odd=list(filter(lambda x : x%2 != 0,Arr1))

print("Count of even no = ",len(even))
print("Count of odd no = ",len(odd))
```

```
Count of even no =  3
Count of odd no =  6
```

In [37]:

```
#10. Write a Python program to add two given lists using map and lambda

list1 = [2,3,4,5,8]
list2 = [7,8,9,10,4]

Add = list(map(lambda x,y : x+y,list1,list2))

print("Addition of the given list = ",Add)
```

```
Addition of the given list =  [9, 11, 13, 15, 12]
```

In [41]:

```
#11. Write a Python program to find numbers divisible by nineteen or thirteen from a list o
#numbers using Lambda

list1=[34,55,52,117,38,58,49,34,39]

result = list(filter(lambda x: x%19==0 or x%13==0,list1))
result
```

Out[41]:

```
[52, 117, 38, 39]
```

In [42]:

```
list1=range(100)

result = list(filter(lambda x: x%19==0 or x%13==0,list1))
result
```

Out[42]:

```
[0, 13, 19, 26, 38, 39, 52, 57, 65, 76, 78, 91, 95]
```

In [44]:

```
#12. Write a Python program to find palindromes in a given list of strings using Lambda

list1 = ["madam","nun","Python","sos","Data"]


result = list(filter(lambda x: x == "".join(reversed(x)),list1))

result
```

Out[44]:

```
['madam', 'nun', 'sos']
```

In [2]:

```
str1 = ['madam','nun','manoj']

result=list(filter(lambda x : x == x[::-1],str1 ))

result
```

Out[2]:

```
['madam', 'nun']
```

In [ ]:

```
#13. Write a Python program to find all anagrams of a string in a given list of strings usi
```

In [10]:

```
list1=range(10)

cube=[(i,i**3) for i in list1]
print(cube)
```

```
[(0, 0), (1, 1), (2, 8), (3, 27), (4, 64), (5, 125), (6, 216), (7, 343), (8,
512), (9, 729)]
```

14. Write a Python program that multiplies each number of a given list with a given number using lambda function. Print the result

In [3]:

```
list1 =[12,13,14,15,16,17,18]
num = 5

def mult(list1,num):
    list2 = []
    for i in list1:
        list2.append(i * num)

    return list2


mult(list1,num)
```

Out[3]:

```
[60, 65, 70, 75, 80, 85, 90]
```

In [4]:

```
list1 =[12,13,14,15,16,17,18]
num = 5

result=list(map(lambda x : x*num, list1))

print(result)
```

```
[60, 65, 70, 75, 80, 85, 90]
```

15. Write a Python program to calculate the sum of the positive and negative numbers of a given list of numbers using lambda function

In [5]:

```
list1 = [2,3,4,5,6,-3,-4,-8,-3]
print("original list",list1)

positive_num = list(filter(lambda x: x > 0,list1))
negative_num = list(filter(lambda x: x < 0,list1))

print("positive_num:>>>",positive_num)
print("negative_num:>>",negative_num)

print("sum of positive numbers:>>",sum(positive_num))
print("sum of positive numbers:>>",sum(negative_num))
```

```
original list [2, 3, 4, 5, 6, -3, -4, -8, -3]
positive_num:>>> [2, 3, 4, 5, 6]
negative_num:>> [-3, -4, -8, -3]
sum of positive numbers:>> 20
sum of positive numbers:>> -18
```

16. Write a Python program to find the list with

maximum and minimum length using lambda

In [6]:

```
def max_length_list(input_list):
    max_length = max(len(x) for x in input_list )
    max_list = max(input_list, key = lambda i: len(i))
    return(max_length, max_list)

def min_length_list(input_list):
    min_length = min(len(x) for x in input_list )
    min_list = min(input_list, key = lambda i: len(i))
    return(min_length, min_list)

list1 = [[0], [1, 3], [5, 7], [9, 11], [13, 15, 17]]

print(list1)
print(max_length_list(list1))
print(min_length_list(list1))
```

```
[[0], [1, 3], [5, 7], [9, 11], [13, 15, 17]]
(3, [13, 15, 17])
(1, [0])
```

17. Write a Python program to check whether a specified list is sorted or not using lambda

In [7]:

```
list1 = [2,4,1,-2,10,-3,0,7,-11]
print("Before sorting:",list1)
print()
list1.sort()
print("After sorting:",list1)
```

```
Before sorting: [2, 4, 1, -2, 10, -3, 0, 7, -11]

After sorting: [-11, -3, -2, 0, 1, 2, 4, 7, 10]
```

In [9]:

```
l1 = [2,4,1,-2,10,-3,0,7,-11]


def sort_list(list1):
    print("List before sorting:",list1)

    if list1 == sorted(list1):
        print("list is sorted",list1)
    else:
        print("not sorted")

sort_list(l1)
```

```
List before sorting: [2, 4, 1, -2, 10, -3, 0, 7, -11]
not sorted
```

In [10]:

```
l1 = [2,4,1,-2,10,-3,0,7,-11]

flag = 0
if(l1 == sorted(l1)):
    flag = 1

if (flag) :
    print ("Yes, List is sorted.")
else :
    print ("No, List is not sorted.")
```

```
No, List is not sorted.
```

18. Write a Python program to remove all elements from a given list present in another list using lambda.

In [13]:

```
list1 = [1,2,3,4,5,6,7,8,9,10]
list2 = [2,4,6,8]
result = list(filter(lambda x: x not in list2,list1))
result
```

Out[13]:

```
[1, 3, 5, 7, 9, 10]
```

In [14]:

```
list1 = [1,2,3,4,5,6,7,8,9,10]

list2 = [2,4,6,8]

final_list = [i for i in list1 if i not in list2]

final_list
```

Out[14]:

```
[1, 3, 5, 7, 9, 10]
```

In [15]:

```
test_list = [1, 3, 4, 6, 7]

remove_list = [3, 6]

print ("The original list is : ", test_list)

print ("The original list is : ",remove_list)

for i in remove_list:
    try:
        test_list.remove(i)
    except ValueError:
        pass
print ("The list after performing remove operation is",test_list)
```

```
The original list is :  [1, 3, 4, 6, 7]
The original list is :  [3, 6]
The list after performing remove operation is [1, 4, 7]
```

19. Write a Python program to convert string element to integer inside a given tuple using lambda

In [16]:

```
t =  (('10', 'zwe', '30'), ('15', 'bcd', '5'), ('57', 'rty', '88'))

result = list(map(lambda x: (int(x[0]),int(x[2])),t))

result
```

Out[16]:

```
[(10, 30), (15, 5), (57, 88)]
```

20. Write a Python program to count the occurrences of the items in a given list using lambda

In [19]:

```
def occur_list(list1,num):
    count = 0
    for i in list1:
        if i == num:
            count +=1
    return count

list1 = [2,2,2,4,4,5,6,7,8,1,1,1,1]

num = int(input("Enter the number:>"))

occur_list(list1,num)
```

```
Enter the number:>2
```

Out[19]:

```
3
```

In [22]:

```
list1 = [2,2,2,4,4,5,6,7,8,1,1,1,1]

print(list1.count(1))
```

```
4
```

In [28]:

```
from collections import Counter

# declaring the list
l = [2,2,2,4,4,5,6,7,8,1,1,1,1]

# driver program
x = 2
d = Counter(l)
print('{} has occurred {} times'.format(x, d[x]))
```

```
2 has occurred 3 times
```

In [27]:

```
l = [2,2,2,4,4,5,6,7,8,1,1,1,1]

occurence = {i: l.count(i) for i in l}
print(occurence.get(2))
```

```
3
```

In [26]:

```
result = dict(map(lambda i : (i, list(list1).count(i)), list1))
result.get(2)
```

Out[26]:

```
3
```

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

In [29]:

```
list1 = list(range(1,11))
list2 = list(range(11,21))
list3 = list(range(21,31))

addition = list(map(lambda x,y,z: x+y+z,list1,list2,list3))

print(addition)
```

```
[33, 36, 39, 42, 45, 48, 51, 54, 57, 60]
```

In [30]:

```
list1 = [1,2,3,4,5]
list2 = [1,2,3,4]
list3 = [1,2,3]

addition = list(map(lambda x,y,z: x+y+z,list1,list2,list3))

print(addition)
```

```
[3, 6, 9]
```

22. Write a Python program to listify the list of given strings individually using Python map

In [31]:

```
name = ['Rohit', 'Bala', 'manoj']
print("Original list: ",name)
print()
result = list(map(list, name))
print(result)
```

```
Original list:  ['Rohit', 'Bala', 'manoj']

[['R', 'o', 'h', 'i', 't'], ['B', 'a', 'l', 'a'], ['m', 'a', 'n', 'o', 'j']]
```

23. Write a Python program to square the elements of a list using map() function

In [32]:

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

square = list(map(lambda x : x**2,list1))

square
```

Out[32]:

```
[1, 4, 9, 16, 25]
```

24. Write a Python program to add two given lists and find the difference between lists. Use map() function

In [33]:

```
list1 = [1,2,3,4,5]
list2 = [11,7,18,41,55]
print("list1",list1)
print("list2",list2)

addition = list(map(lambda x,y : x+y,list1,list2))

print("Addition of two list is: ",addition)

difference = list(map(lambda x,y:x-y,list1,list2))

print("Difference of two list is: ",difference)
```

```
list1 [1, 2, 3, 4, 5]
list2 [11, 7, 18, 41, 55]
Addition of two list is:  [12, 9, 21, 45, 60]
Difference of two list is:  [-10, -5, -15, -37, -50]
```

25. Write a Python program to convert a given list of integers and a tuple of integers in a list of strings¶

In [34]:

```
list1 = [2,4,5,6,8]
tuple1 = (4,5,8,7,3)

l1=list(tuple1)
list1.extend(l1)
result=[]
for i in list1:
    result.append(str(i))

print(result)
```

```
['2', '4', '5', '6', '8', '4', '5', '8', '7', '3']
```

26. Write a Python program to compute the sum of elements of an given array of integers, use map() function

In [35]:

```
from array import array
def array_sum(nums_arr):
    sum_n = 0
    for n in nums_arr:
        sum_n += n
    return sum_n

nums = array('i', [1, 2,4,8,11])
print("Original array:",nums)
nums_arr = list(map(int, nums))
result = array_sum(nums_arr)
print("Sum of all elements of the said array:")
print(result)
```

```
Original array: array('i', [1, 2, 4, 8, 11])
Sum of all elements of the said array:
26
```

27. Write a Python program to count the same pair in two given lists. use map() function

In [37]:

```
list1 = [1,2,3,4,5,6]
list2 = [2,3,4,8,9,10]

same = list(map(lambda x,y: x ==y ,list1,list2))

same
```

Out[37]:

```
[False, False, False, False, False, False]
```

In [38]:

```
list1 = [1,2,3,4,5,6]
list2 = [2,3,4,8,9,10]

same =[i for i in list1 if i in list2]
same
```

Out[38]:

```
[2, 3, 4]
```

In [39]:

```
from operator import eq
def count_same_pair(nums1, nums2):
    result = sum(map(eq, nums1, nums2))
    return result

nums1 = [1,2,3,4,5,6,7,8]
nums2 = [2,2,3,1,2,6,7,9]
print("Original lists:")
print(nums1)
print(nums2)
print("\nNumber of same pair of the said two given lists:")
print(count_same_pair(nums1, nums2))
```

```
Original lists:
[1, 2, 3, 4, 5, 6, 7, 8]
[2, 2, 3, 1, 2, 6, 7, 9]

Number of same pair of the said two given lists:
4
```

28. Write a Python program to convert a given list of strings into list of lists using map function

In [41]:

```
str1 = ["man","dog","god","cat"]

def strings_to_listOflist(str1):
    result = map(list, str1)
    return list(result)



print("print the original list:>>",str1)


print(strings_to_listOflist(str1))
```

```
print the original list:>> ['man', 'dog', 'god', 'cat']
[['m', 'a', 'n'], ['d', 'o', 'g'], ['g', 'o', 'd'], ['c', 'a', 't']]
```

29. Write a Python program to convert a given list of tuples to a list of strings using map function

In [42]:

```
list_tuple = [("dog","cat"),("lion","tiger"),("fish","shark")]
print("Original list of tuples:-\n")
print(list_tuple,"\n")

list_str =  list(map(' '.join, list_tuple))
print("List of strings:\n")
print(list_str)
```

```
Original list of tuples:-

[('dog', 'cat'), ('lion', 'tiger'), ('fish', 'shark')]

List of strings:

['dog cat', 'lion tiger', 'fish shark']
```

30. Python program to find the diff. between two lists using filter() function

In [43]:

```
list1 = [1,2,10,14,5,88,74,6]
list2 = [2,4,5,6,88,78]


result = list(filter(lambda x: x not in list1,list2))
result1 =list(filter(lambda x: x not in list2,list1))
print( result + result1 )
```

```
[4, 78, 1, 10, 14, 74]
```

In [44]:

```
list1 = [1,2,10,14,5,88,74,6]
list2 = [2,4,5,6,88,78]

set_dif = set(list1).symmetric_difference(set(list2))
temp3 = list(set_dif)
print(temp3)
```

```
[1, 4, 74, 10, 14, 78]
```

31. Python program to remove stop words from string using filter() function

In [45]:

```
str1 = "cat dog mat stop bat tat"
str1_str = str1.split()
print(str1_str)

def remove_word(str1):
    list1 = str1.split()
    for i in list1:
        if i == 'stop':
            list1.remove('stop')
    return " ".join(list1)

str1 = "cat dog mat stop bat tat"
remove_word(str1)
```

```
['cat', 'dog', 'mat', 'stop', 'bat', 'tat']
```

Out[45]:

```
'cat dog mat bat tat'
```

In [46]:

```
str1 = "cat dog mat stop bat tat"
str2 =str1.split()
result=list(filter(lambda x:x!="stop",str2))
print(" ".join(result))
```

```
cat dog mat bat tat
```

32. Python program to find common items in two arrays using lambda and filter() function

In [47]:

```
l1 = [1,2,3,4,5,6,7]
l2 = [5,8,2,9,1]
s1 = set(l1)
s2 = set(l2)
result = s1.intersection(s2)
result
```

Out[47]:

```
{1, 2, 5}
```

In [49]:

```
l1 = [1,2,3,4,5,6,7]
l2 = [5,8,2,9,1]

result = list(filter(lambda x: x in l1,l2))

result
```

Out[49]:

```
[5, 2, 1]
```

33. Python program to filter odd numbers from the list using filter() function

In [50]:

```
list1 =[2,7,8,9,10,24,67,48,15,77,37,13]

result = list(filter(lambda x:x%2!=0,list1))

result
```

Out[50]:

```
[7, 9, 67, 15, 77, 37, 13]
```

34. Python program to filter even numbers from the list using filter() function

In [51]:

```
list1 =[2,7,8,9,10,24,67,48,15,77,37,13]

result = list(filter(lambda x:x%2==0,list1))

result
```

Out[51]:

```
[2, 8, 10, 24, 48]
```

35. Python program that filters non-vowels from the list using filter() function¶

In [52]:

```
str1 =['a','b','c','d','e','f','g','h','h','i','j','k','l','m','n','o','p']

vowels = ['a','e','i','o','u']
result=list(filter(lambda x:x not in vowels,str1))
print(result)
```

```
['b', 'c', 'd', 'f', 'g', 'h', 'h', 'j', 'k', 'l', 'm', 'n', 'p']
```

In [ ]:

```

```