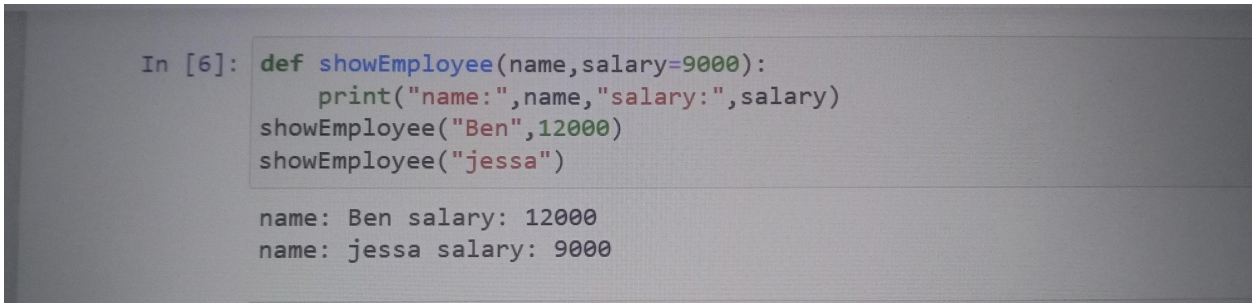
TASK 4

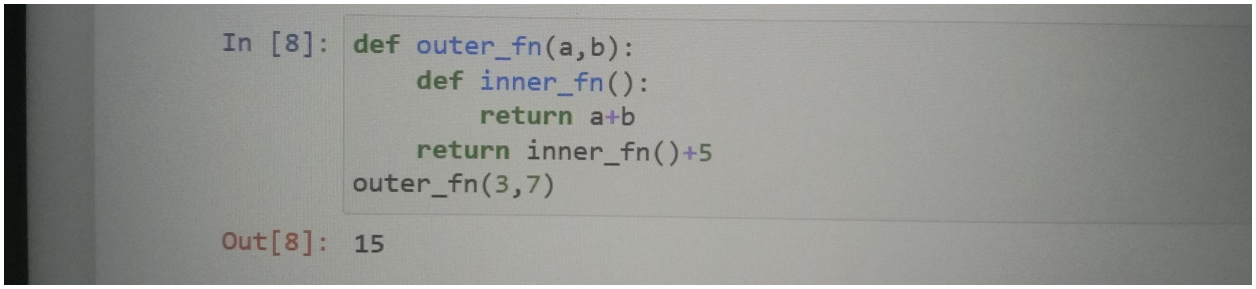
Exercise 1: Create a function with a default argument
Write a program to create a function show_employee() using the following conditions.
It should accept the employee's name and salary and display both.
If the salary is missing in the function call then assign default value 9000 to salary
Given:
showEmployee("Ben", 12000)
showEmployee("Jessa")
Expected output:
Name: Ben salary: 12000
Name: Jessa salary: 9000



```
In [6]: def showEmployee(name,salary=9000):
            print("name:",name,"salary:",salary)
        showEmployee("Ben",12000)
        showEmployee("jessa")

        name: Ben salary: 12000
        name: jessa salary: 9000
```

Exercise 2: Create an inner function to calculate the addition in the following way
Create an outer function that will accept two parameters, a and b
Create an inner function inside an outer function that will calculate the addition of a and b
At last, an outer function will add 5 into addition and return it



```
In [8]: def outer_fn(a,b):
            def inner_fn():
                return a+b
            return inner_fn()+5
        outer_fn(3,7)

Out[8]: 15
```

Exercise 3: Generate a Python list of all the even numbers between 4 to 30

```
In [11]: def even_list():
             l1=[]
             for x in range(4,31):
                 if x%2==0:
                     l1.append(x)
             print(l1)
         even_list()
```

```
[4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30]
```

Exercise 4: Lambda Function to Check if value is in a List
Given a list, the task is to write a Python program to check if the value exists in the list or not using the lambda function.
Input : L = [1, 2, 3, 4, 5]
element = 4
Output : Element is Present in the list

Input : L = [1, 2, 3, 4, 5]
element = 8
Output : Element is NOT Present in the list

```
In [29]: L=[1,2,3,4,5]
         a=4
         x=lambda k:k in L
         print("element is present" if x(a) else "element is not present")
         a=8
         x=lambda k:k in L
         print("element is present" if x(a) else "element is not present")
```

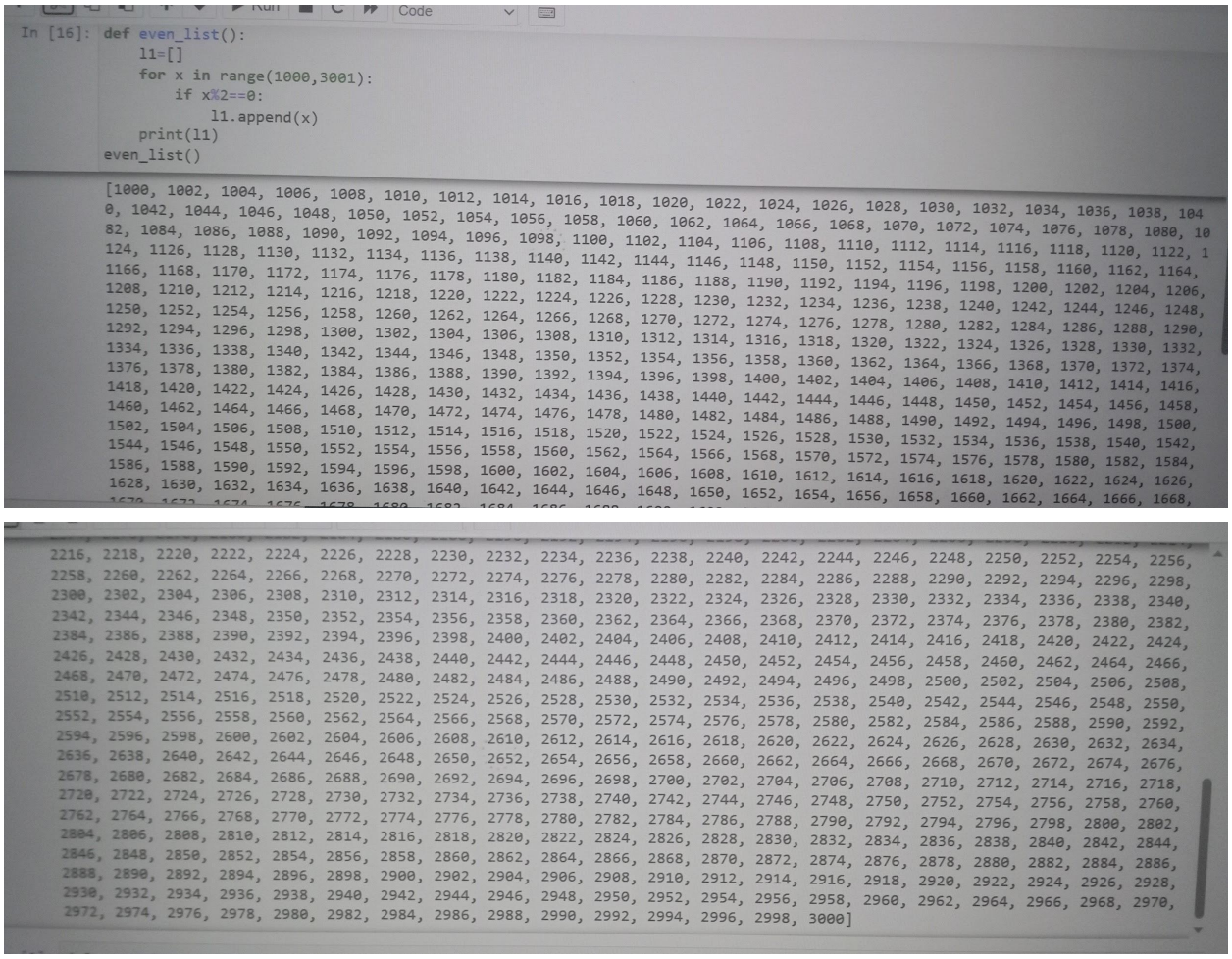
```
element is present
element is not present
```

Exercise 5: Sort list of tuples with their sum
Sort the points based on their sum of elements in the tuples
points = [(1, 2), (5, 3), (0, 7), (3, 1)]

```
In [2]: a=[(1,2),(5,3),(0,7),(3,1)]
        a.sort(key=lambda x:x[0]+x[1])
        print(a)

        [(1, 2), (3, 1), (0, 7), (5, 3)]
```

Exercise 6 :
Write a python function, which will find all such numbers between 1000 and 3000 (both included) such that each digit of the number is an even number. Return the results as a list



```
In [16]: def even_list():
             l1=[]
             for x in range(1000,3001):
                 if x%2==0:
                     l1.append(x)
             print(l1)
         even_list()

[1000, 1002, 1004, 1006, 1008, 1010, 1012, 1014, 1016, 1018, 1020, 1022, 1024, 1026, 1028, 1030, 1032, 1034, 1036, 1038, 104
0, 1042, 1044, 1046, 1048, 1050, 1052, 1054, 1056, 1058, 1060, 1062, 1064, 1066, 1068, 1070, 1072, 1074, 1076, 1078, 1080, 10
82, 1084, 1086, 1088, 1090, 1092, 1094, 1096, 1098, 1100, 1102, 1104, 1106, 1108, 1110, 1112, 1114, 1116, 1118, 1120, 1122, 1
124, 1126, 1128, 1130, 1132, 1134, 1136, 1138, 1140, 1142, 1144, 1146, 1148, 1150, 1152, 1154, 1156, 1158, 1160, 1162, 1164,
1166, 1168, 1170, 1172, 1174, 1176, 1178, 1180, 1182, 1184, 1186, 1188, 1190, 1192, 1194, 1196, 1198, 1200, 1202, 1204, 1206,
1208, 1210, 1212, 1214, 1216, 1218, 1220, 1222, 1224, 1226, 1228, 1230, 1232, 1234, 1236, 1238, 1240, 1242, 1244, 1246, 1248,
1250, 1252, 1254, 1256, 1258, 1260, 1262, 1264, 1266, 1268, 1270, 1272, 1274, 1276, 1278, 1280, 1282, 1284, 1286, 1288, 1290,
1292, 1294, 1296, 1298, 1300, 1302, 1304, 1306, 1308, 1310, 1312, 1314, 1316, 1318, 1320, 1322, 1324, 1326, 1328, 1330, 1332,
1334, 1336, 1338, 1340, 1342, 1344, 1346, 1348, 1350, 1352, 1354, 1356, 1358, 1360, 1362, 1364, 1366, 1368, 1370, 1372, 1374,
1376, 1378, 1380, 1382, 1384, 1386, 1388, 1390, 1392, 1394, 1396, 1398, 1400, 1402, 1404, 1406, 1408, 1410, 1412, 1414, 1416,
1418, 1420, 1422, 1424, 1426, 1428, 1430, 1432, 1434, 1436, 1438, 1440, 1442, 1444, 1446, 1448, 1450, 1452, 1454, 1456, 1458,
1460, 1462, 1464, 1466, 1468, 1470, 1472, 1474, 1476, 1478, 1480, 1482, 1484, 1486, 1488, 1490, 1492, 1494, 1496, 1498, 1500,
1502, 1504, 1506, 1508, 1510, 1512, 1514, 1516, 1518, 1520, 1522, 1524, 1526, 1528, 1530, 1532, 1534, 1536, 1538, 1540, 1542,
1544, 1546, 1548, 1550, 1552, 1554, 1556, 1558, 1560, 1562, 1564, 1566, 1568, 1570, 1572, 1574, 1576, 1578, 1580, 1582, 1584,
1586, 1588, 1590, 1592, 1594, 1596, 1598, 1600, 1602, 1604, 1606, 1608, 1610, 1612, 1614, 1616, 1618, 1620, 1622, 1624, 1626,
1628, 1630, 1632, 1634, 1636, 1638, 1640, 1642, 1644, 1646, 1648, 1650, 1652, 1654, 1656, 1658, 1660, 1662, 1664, 1666, 1668,
```



```
2216, 2218, 2220, 2222, 2224, 2226, 2228, 2230, 2232, 2234, 2236, 2238, 2240, 2242, 2244, 2246, 2248, 2250, 2252, 2254, 2256,
2258, 2260, 2262, 2264, 2266, 2268, 2270, 2272, 2274, 2276, 2278, 2280, 2282, 2284, 2286, 2288, 2290, 2292, 2294, 2296, 2298,
2300, 2302, 2304, 2306, 2308, 2310, 2312, 2314, 2316, 2318, 2320, 2322, 2324, 2326, 2328, 2330, 2332, 2334, 2336, 2338, 2340,
2342, 2344, 2346, 2348, 2350, 2352, 2354, 2356, 2358, 2360, 2362, 2364, 2366, 2368, 2370, 2372, 2374, 2376, 2378, 2380, 2382,
2384, 2386, 2388, 2390, 2392, 2394, 2396, 2398, 2400, 2402, 2404, 2406, 2408, 2410, 2412, 2414, 2416, 2418, 2420, 2422, 2424,
2426, 2428, 2430, 2432, 2434, 2436, 2438, 2440, 2442, 2444, 2446, 2448, 2450, 2452, 2454, 2456, 2458, 2460, 2462, 2464, 2466,
2468, 2470, 2472, 2474, 2476, 2478, 2480, 2482, 2484, 2486, 2488, 2490, 2492, 2494, 2496, 2498, 2500, 2502, 2504, 2506, 2508,
2510, 2512, 2514, 2516, 2518, 2520, 2522, 2524, 2526, 2528, 2530, 2532, 2534, 2536, 2538, 2540, 2542, 2544, 2546, 2548, 2550,
2552, 2554, 2556, 2558, 2560, 2562, 2564, 2566, 2568, 2570, 2572, 2574, 2576, 2578, 2580, 2582, 2584, 2586, 2588, 2590, 2592,
2594, 2596, 2598, 2600, 2602, 2604, 2606, 2608, 2610, 2612, 2614, 2616, 2618, 2620, 2622, 2624, 2626, 2628, 2630, 2632, 2634,
2636, 2638, 2640, 2642, 2644, 2646, 2648, 2650, 2652, 2654, 2656, 2658, 2660, 2662, 2664, 2666, 2668, 2670, 2672, 2674, 2676,
2678, 2680, 2682, 2684, 2686, 2688, 2690, 2692, 2694, 2696, 2698, 2700, 2702, 2704, 2706, 2708, 2710, 2712, 2714, 2716, 2718,
2720, 2722, 2724, 2726, 2728, 2730, 2732, 2734, 2736, 2738, 2740, 2742, 2744, 2746, 2748, 2750, 2752, 2754, 2756, 2758, 2760,
2762, 2764, 2766, 2768, 2770, 2772, 2774, 2776, 2778, 2780, 2782, 2784, 2786, 2788, 2790, 2792, 2794, 2796, 2798, 2800, 2802,
2804, 2806, 2808, 2810, 2812, 2814, 2816, 2818, 2820, 2822, 2824, 2826, 2828, 2830, 2832, 2834, 2836, 2838, 2840, 2842, 2844,
2846, 2848, 2850, 2852, 2854, 2856, 2858, 2860, 2862, 2864, 2866, 2868, 2870, 2872, 2874, 2876, 2878, 2880, 2882, 2884, 2886,
2888, 2890, 2892, 2894, 2896, 2898, 2900, 2902, 2904, 2906, 2908, 2910, 2912, 2914, 2916, 2918, 2920, 2922, 2924, 2926, 2928,
2930, 2932, 2934, 2936, 2938, 2940, 2942, 2944, 2946, 2948, 2950, 2952, 2954, 2956, 2958, 2960, 2962, 2964, 2966, 2968, 2970,
2972, 2974, 2976, 2978, 2980, 2982, 2984, 2986, 2988, 2990, 2992, 2994, 2996, 2998, 3000]
```

Exercise 7 :
Write a python function that accepts a sentence and calculate and return 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

```
In [1]: def count_letter_digits(sentence):
            letters=0
            digits=0
            for char in sentence:
                if char.isalpha():
                    letters +=1
                elif char.isdigit():
                    digits +=1
            return(letters,digits)
        sentence="hello world!123"
        result=count_letter_digits(sentence)
        print(f"LETTERS  {result[0]}")
        print(f"DIGITS {result[1]}")

        LETTERS  10
        DIGITS 3
```

Exercise 8 MAP:

Write a Python program to convert all the characters into uppercase and lowercase and eliminate duplicate letters from a given sequence. Use the map() function

```
In [14]: original="There Lives a Devil"
         upp=map(str.upper,original)
         low=map(str.lower,original)
         unique_upp={x for x in upp}
         unique_low={x for x in low}
         print("unique uppercase:",unique_upp)
         print("unique lowercase",unique_low)

         unique uppercase: {'T', 'R', 'E', 'V', 'S', 'H', 'A', 'D', 'L', 'I', ' '}
         unique lowercase {'t', 'l', 'e', 'a', 's', 'h', 'v', 'i', 'd', 'r', ' '}
```

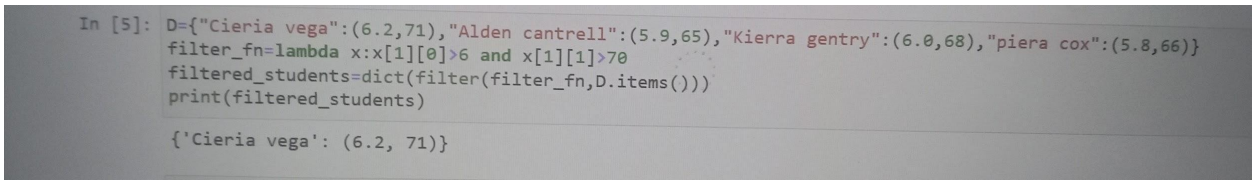
Exercise 9 MAP:

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

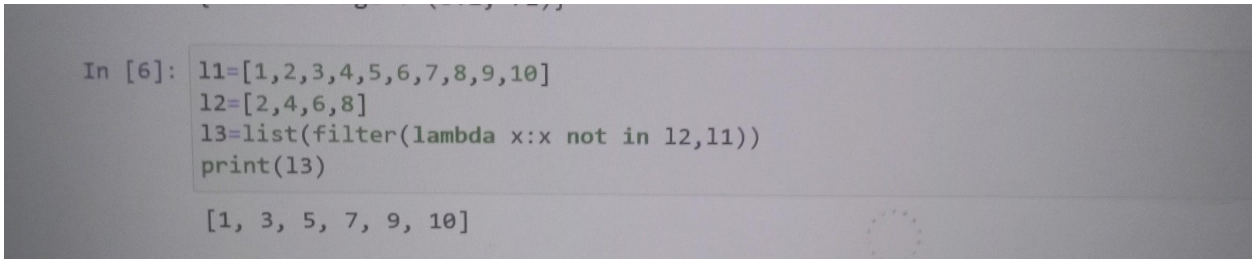
```
In [2]: l1=[4,9,8,7,6]
        l2=[3,7,2,5,1]
        add=list(map(lambda x,y:x+y,l1,l2))
        print("added l1 and l2:",add)
        di=list(map(lambda x,y:x-y,l1,l2))
        print("the diffrence of l1 and l2:",di)

        added l1 and l2: [7, 16, 10, 12, 7]
        the diffrence of l1 and l2: [1, 2, 6, 2, 5]
```

Exercise 9 Filter:
Write a Python program to filter the height and weight of students, which are stored in a dictionary using lambda.
Original Dictionary:
{'Cierra Vega': (6.2, 71), 'Alden Cantrell': (5.9, 65), 'Kierra Gentry': (6.0, 68), 'Pierre Cox': (5.8, 66)}
Height> 6ft and Weight> 70kg:
{'Cierra Vega': (6.2, 71)}



Exercise 10 Filter:
Write a Python program to remove all elements from a given list present in another list using lambda.
Original lists:
list1: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
list2: [2, 4, 6, 8]
Remove all elements from 'list1' present in 'list2':
[1, 3, 5, 7, 9, 10]



Exercise 11 Reduce:
Write a Python program to calculate the product of a given list of numbers using lambda.
list1: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
Product of the said list numbers:
3628800

```
In [7]: import functools
        l1=[1,2,3,4,5,6,7,8,9,10]
        res=functools.reduce(lambda x,y:x*y,l1)
        print(res)

        3628800
```

Exercise 12 Reduce:
Write a Python program to multiply all the numbers in a given list using lambda.
Original list:
[4, 3, 2, 2, -1, 18]
Multiply all the numbers of the said list: -864

```
In [8]: import functools
        l1=[4,3,2,2,-1,18]
        res=functools.reduce(lambda x,y:x*y,l1)
        print(res)

        -864
```

Exercise 13 Reduce:
Write a Python program to calculate the average value of the numbers in a given tuple of tuples using lambda.
Original Tuple:
((10, 10, 10), (30, 45, 56), (81, 80, 39), (1, 2, 3))
Average value of the numbers of the said tuple of tuples:
(30.5, 34.25, 27.0)

```
In [21]: from functools import reduce
         l1=((10,10,10),(30,45,56),(81,80,39),(1,2,3))
         res=tuple(reduce(lambda x,y:map(sum,zip(x,y)),l1))
         print(f"the avarage: {tuple(map(lambda x:x/len(l1),res))}")

         the avarage: (30.5, 34.25, 27.0)
```

Exercise 13:
Write a Python program to sort a given mixed list of integers and strings using lambda. Numbers must be sorted before strings.
Original list:
[19, 'red', 12, 'green', 'blue', 10, 'white', 'green', 1]
Sort the said mixed list of integers and strings:
[1, 10, 12, 19, 'blue', 'green', 'green', 'red', 'white']

```
In [9]: og_list=[19,'red','green','blue',10,'white','green',1]
        result=sorted(og_list,key=lambda x:(str(x),x))
        print(result)

        [1, 10, 19, 'blue', 'green', 'green', 'red', 'white']
```

Exercise 14:
Write a Python program to count the occurrences of items in a given list using lambda.
Original list:
[3, 4, 5, 8, 0, 3, 8, 5, 0, 3, 1, 5, 2, 3, 4, 2]
Count the occurrences of the items in the said list:
{3: 4, 4: 2, 5: 3, 8: 2, 0: 2, 1: 1, 2: 2}

```
In [8]: og_list=[3,4,5,8,0,3,8,5,0,3,1,52,3,4,2]
        result={x:og_list.count(x) for x in og_list}
        print(result)

        {3: 4, 4: 2, 5: 2, 8: 2, 0: 2, 1: 1, 52: 1, 2: 1}
```

Exercise 15:
Write a Python program to remove None values from a given list using the lambda function.
Original list:
[12, 0, None, 23, None, -55, 234, 89, None, 0, 6, -12]
Remove None value from the said list:
[12, 0, 23, -55, 234, 89, 0, 6, -12]

```
In [7]: og_list=[12,0,None,23,None,-55,234,89,None,0,6,12]
        required_list=filter(lambda x:x is not None, og_list)
        list(required_list)

Out[7]: [12, 0, 23, -55, 234, 89, 0, 6, 12]
```