

## 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
```

Task4.1.py > show\_employee

```
1 def show_employee(name,salary=9000):  
2     print("Employee name: ", name)  
3     print("Salary: ", str(salary))  
4  
5 show_employee("Ben",12000)  
6 show_employee("Jessica")
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

JUPYTER

```
PS C:\Users\aishw\entri_python\Python Assignments> python Task4.1.py  
Employee name: Ben  
Salary: 12000  
Employee name: Jessica  
Salary: 9000  
PS C:\Users\aishw\entri_python\Python Assignments> 
```

## 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

```
Task4.2.py > ...
1  def sum(a,b):
2      def def_add():
3          add = a+b
4          return add
5      add = def_add()
6      return add + 5
7  print(sum(3,5))
```

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   JUPYTER

```
PS C:\Users\aisw\entri_python\Python Assignments> python Task4.2.py
13
PS C:\Users\aisw\entri_python\Python Assignments> 
```

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

```
Task4.3.py > ...
1  even = []
2  for i in range(4,30) :
3      if(i%2 == 0):
4          even.append(i)
5
6  print(even)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER powershell +

```
PS C:\Users\aisw\entri_python\Python Assignments> python Task4.3.py
[4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28]
PS C:\Users\aisw\entri_python\Python Assignments> |
```

## 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

```
Task4.4.py > ...
1  lists = [2,3,5,2,4,44]
2  ele = 6
3  x = lambda L , ele : True if ele in L else False
4  if (x(lists , ele)) :
5      print("Present")
6  else :
7      print("Not present")
8

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  JUPYTER

PS C:\Users\aishw\entri_python\Python Assignments> python Task4.3.py
[4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28]
PS C:\Users\aishw\entri_python\Python Assignments> python Task4.3.py
[4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28]
PS C:\Users\aishw\entri_python\Python Assignments> python Task4.4.py
Present
PS C:\Users\aishw\entri_python\Python Assignments> python Task4.4.py
Not present
PS C:\Users\aishw\entri_python\Python Assignments> █
```

## 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)]
```

```
Task4.5.py > ...
1  lists = [(3,4),(2,3), (4,5),(44,55)]
2
3  lists.sort(key = lambda x :x[0] + x[1])
4
5  print(lists)
```

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   JUPYTER

```
PS C:\Users\aisw\entri_python\Python Assignments> python Task4.5.py
[(2, 3), (3, 4), (4, 5), (44, 55)]
PS C:\Users\aisw\entri_python\Python Assignments> 
```

## 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

```
Task4.6.py
1  l = []
2  for i in range(1000,3001):
3      if(i%2 == 0):
4          l.append(i)
5
6  print(l)
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** JUPYTER powershell + - [ ] [X] ... ^ X

```
[(2, 3), (3, 4), (4, 5), (44, 55)]
PS C:\Users\aisw\entri_python\Python Assignments> python Task4.6.py
[1000, 1002, 1004, 1006, 1008, 1010, 1012, 1014, 1016, 1018, 1020, 1022, 1024, 1026, 1028, 1030, 1032, 1034, 1036, 1038, 1040,
1042, 1044, 1046, 1048, 1050, 1052, 1054, 1056, 1058, 1060, 1062, 1064, 1066, 1068, 1070, 1072, 1074, 1076, 1078, 1080, 1082, 1
084, 1086, 1088, 1090, 1092, 1094, 1096, 1098, 1100, 1102, 1104, 1106, 1108, 1110, 1112, 1114, 1116, 1118, 1120, 1122, 1124, 11
26, 1128, 1130, 1132, 1134, 1136, 1138, 1140, 1142, 1144, 1146, 1148, 1150, 1152, 1154, 1156, 1158, 1160, 1162, 1164, 1166, 116
8, 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, 1
338, 1340, 1342, 1344, 1346, 1348, 1350, 1352, 1354, 1356, 1358, 1360, 1362, 1364, 1366, 1368, 1370, 1372, 1374, 1376, 1378, 13
80, 1382, 1384, 1386, 1388, 1390, 1392, 1394, 1396, 1398, 1400, 1402, 1404, 1406, 1408, 1410, 1412, 1414, 1416, 1418, 1420, 142
2, 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, 1
592, 1594, 1596, 1598, 1600, 1602, 1604, 1606, 1608, 1610, 1612, 1614, 1616, 1618, 1620, 1622, 1624, 1626, 1628, 1630, 1632, 16
34, 1636, 1638, 1640, 1642, 1644, 1646, 1648, 1650, 1652, 1654, 1656, 1658, 1660, 1662, 1664, 1666, 1668, 1670, 1672, 1674, 167
6, 1678, 1680, 1682, 1684, 1686, 1688, 1690, 1692, 1694, 1696, 1698, 1700, 1702, 1704, 1706, 1708, 1710, 1712, 1714, 1716, 1718
, 1720, 1722, 1724, 1726, 1728, 1730, 1732, 1734, 1736, 1738, 1740, 1742, 1744, 1746, 1748, 1750, 1752, 1754, 1756, 1758, 1760,
1762, 1764, 1766, 1768, 1770, 1772, 1774, 1776, 1778, 1780, 1782, 1784, 1786, 1788, 1790, 1792, 1794, 1796, 1798, 1800, 1802,
1804, 1806, 1808, 1810, 1812, 1814, 1816, 1818, 1820, 1822, 1824, 1826, 1828, 1830, 1832, 1834, 1836, 1838, 1840, 1842, 1844, 1
846, 1848, 1850, 1852, 1854, 1856, 1858, 1860, 1862, 1864, 1866, 1868, 1870, 1872, 1874, 1876, 1878, 1880, 1882, 1884, 1886, 1888, 1890, 1892, 1894, 1896, 1898, 1900, 1902, 1904, 1906, 1908, 1910, 1912, 1914, 1916, 1918, 1920, 1922, 1924, 1926, 1928, 1930, 1932, 1934, 1936, 1938, 1940, 1942, 1944, 1946, 1948, 1950, 1952, 1954, 1956, 1958, 1960, 1962, 1964, 1966, 1968, 1970, 1972, 1974, 1976, 1978, 1980, 1982, 1984, 1986, 1988, 1990, 1992, 1994, 1996, 1998, 2000, 2002, 2004, 2006, 2008, 2010, 2012, 2014, 2016, 2018, 2020, 2022, 2024, 2026, 2028, 2030, 2032, 2034, 2036, 2038, 2040, 2042, 2044, 2046, 2048, 2050, 2052, 2054, 2056, 2058, 2060, 2062, 2064, 2066, 2068, 2070, 2072, 2074, 2076, 2078, 2080, 2082, 2084, 2086, 2088, 2090, 2092, 2094, 2096, 2098, 2100, 2102, 2104, 2106, 2108, 2110, 2112, 2114, 2116, 2118, 2120, 2122, 2124, 2126, 2128, 2130, 2132, 2134, 2136, 2138, 2140, 2142, 2144, 2146, 2148, 2150, 2152, 2154, 2156, 2158, 2160, 2162, 2164, 2166, 2168, 2170, 2172, 2174, 2176, 2178, 2180, 2182, 2184, 2186, 2188, 2190, 2192, 2194, 2196, 2198, 2200, 2202, 2204, 2206, 2208, 2210, 2212, 2214, 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

```
Task4.7.py > ...
1  s = input("Enter a string: ")
2  d = 0
3  a = 0
4  for i in s :
5      if(i.isdigit()):
6          d = d+1
7      elif(i.isalpha()):
8          a = a+1
9  print("Digit",d)
10 print("Alphabets", a)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

```
PS C:\Users\aisw\entri_python\Python Assignments> python Task4.7.py
Enter a string: rerww rqwe 32 22
Digit 4
Alphabets 9
PS C:\Users\aisw\entri_python\Python Assignments> |
```

Ln 10, Col 22 Spaces: 4

## 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



```
Task4.8.py > ...
1  s = input("Enter a string: ")
2
3  u = set(s.upper())
4  l = set(s.lower())
5  print("Upper :", u)
6  print("Lower :", l)
7  res = map(u,s)
8  res2 = map(l,s)
9  print(res)
10 print(res2)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER powershell + v

```
PS C:\Users\aislw\entri_python\Python Assignments> python Task4.8.py
Enter a string: Hello Good Morning
Upper : {'E', ' ', 'M', 'H', 'G', 'N', 'R', 'I', 'O', 'D', 'L'}
Lower : {'h', 'n', ' ', 'r', 'l', 'i', 'g', 'm', 'd', 'e', 'o'}
<map object at 0x000002B709591750>
<map object at 0x000002B7095917B0>
PS C:\Users\aislw\entri_python\Python Assignments> |
```

## Exercise 9 MAP:

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

```
Task4.9.py > ...
1  l1 = [3,2,4,2,4,2,5,2,4]
2  l2 = [4,3,4,3,5,3]
3
4  l = l1+l2
5  m = [x for x in l1 if x not in l2]
6  x = map(l,l1,l2)
7  print(x)
8  print("Added: ",l)
9  y = map(l,l1,l2)
10 print(y)
11 print("Difference: ", m)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

```
PS C:\Users\aisw\entri_python\Python Assignments> python Task4.9.py
Traceback (most recent call last):
  File "C:\Users\aisw\entri_python\Python Assignments\Task4.9.py", line 5, in <module>
    m = l1-l2
    ~~~~
TypeError: unsupported operand type(s) for -: 'list' and 'list'
PS C:\Users\aisw\entri_python\Python Assignments> python Task4.9.py
<map object at 0x00000219600117E0>
Added:  [3, 2, 4, 2, 4, 2, 5, 2, 4, 4, 3, 4, 3, 5, 3]
<map object at 0x0000021960011870>
Difference:  [2, 2, 2, 2]
PS C:\Users\aisw\entri_python\Python Assignments>
```

## Exercise 10 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)}

```
Task4.10.py > ...
1  def fil(students):
2      result = dict(filter(lambda x: (x[1][0], x[1][1]) > (6.0, 70), students.items()))
3      return result
4  students = {'Cierra Vega': (6.2, 70), 'Alden Cantrell': (5.9, 65), 'Kierra Gentry': (6.0, 68), 'Pierre Cox': (5.8, 66)}
5  print("Dictionary:", students)
6
7  print("Height> 6ft and Weight> 70kg:")
8  print(fil(students))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

PS C:\Users\aisw\entri\_python\Python Assignments> python Task4.10.py  
Dictionary: {'Cierra Vega': (6.2, 70), 'Alden Cantrell': (5.9, 65), 'Kierra Gentry': (6.0, 68), 'Pierre Cox': (5.8, 66)}  
Height> 6ft and Weight> 70kg:  
{'Cierra Vega': (6.2, 70)}  
PS C:\Users\aisw\entri\_python\Python Assignments> |

## Exercise 11 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]

```
Task4.11.py > ...
1  def filter_list(list1, list2):
2      result = list(filter(lambda x: x not in list2, list1))
3      return result
4  list1 = [3,4,2,1,5,6]
5  list2 = [3,4,11,22]
6  print("Original lists:")
7  print("list1:", list1)
8  print("list2:", list2)
9
10 print("Result: " ,filter_list(list1, list2))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER powershell

```
PS C:\Users\aisw\entri_python\Python Assignments> python Task4.10.py
Dictionary: {'Cierra Vega': (6.2, 70), 'Alden Cantrell': (5.9, 65), 'Kierra Gentry': (6.0, 68), 'Pierre Cox'
Height> 6ft and Weight> 70kg:
{'Cierra Vega': (6.2, 70)}
PS C:\Users\aisw\entri_python\Python Assignments> python Task4.11.py
Original lists:
list1: [3, 4, 2, 1, 5, 6]
list2: [3, 4, 11, 22]
Result: [2, 1, 5, 6]
PS C:\Users\aisw\entri_python\Python Assignments> 
```

## Exercise 12 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

```
Task4.12.py > ...
1 import functools
2 def remove_dup(nums):
3     result = functools.reduce(lambda x, y: x * y, nums, 1)
4     return result
5 l1 = [1,2,3,4,5,6,7,8,9,10]
6 l2 = [2.2,4.12,6.6,8.1,8.3]
7 print("list1:", l1)
8 print("Product of the said list numbers:")
9 print(remove_dup(l1))
10 print("list2:", l2)
11 print("Product of the said list numbers:")
12 print(remove_dup(l2))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER powershell +

```
PS C:\Users\aisw\entri_python\Python Assignments> python Task4.10.py
Dictionary: {'Cierra Vega': (6.2, 70), 'Alden Cantrell': (5.9, 65), 'Kierra Gentry': (6.0, 68), 'Pierre Cox':
Height> 6ft and Weight> 70kg:
{'Cierra Vega': (6.2, 70)}
PS C:\Users\aisw\entri_python\Python Assignments> python Task4.11.py
Original lists:
list1: [3, 4, 2, 1, 5, 6]
list2: [3, 4, 11, 22]
Result: [2, 1, 5, 6]
PS C:\Users\aisw\entri_python\Python Assignments> python Task4.12.py
list1: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
Product of the said list numbers:
3628800
list2: [2.2, 4.12, 6.6, 8.1, 8.3]
Product of the said list numbers:
4021.8599520000007
PS C:\Users\aisw\entri_python\Python Assignments>
```

## Exercise 13 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

```
Task4.13.py > ...
1  import functools
2  l = [4,3,2,2,-1,18]
3
4  p = functools.reduce(lambda x , y : x*y , l)
5
6  print("Product : ", p)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

```
PS C:\Users\aiashw\entri_python\Python Assignments> python Task4.13.py
Product : -864
PS C:\Users\aiashw\entri_python\Python Assignments> 
```

## Exercise 14 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)

```
Task4.14.py > ...
1  from functools import reduce
2  t=((10, 10, 10), (30, 45, 56), (81, 80, 39), (1, 2, 3))
3
4  result = tuple(reduce(lambda x, y: map(sum, zip(x,y)), t))
5  print("Average value of tuples:" ,tuple(map(lambda x: x/len(t), result)))
6
7

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  JUPYTER  powershell +
PS C:\Users\aisw\entri_python\Python Assignments> python Task4.14.py
Average value of tuples: {(30.5, 34.25, 27.0)}
PS C:\Users\aisw\entri_python\Python Assignments> 
```

## Exercise 15:

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']

```
Task4.15.py > ...
1  def sort_mixed_list(mixed_list):
2      mixed_list.sort(key=lambda e: (isinstance(e, str), e))
3      return mixed_list
4  list = [19, 'red', 12, 'green', 'blue', 10, 'white', 'green', 1]
5  print("Original list:", list)
6
7  print("Result: ", sort_mixed_list(list))
```

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   JUPYTER

```
PS C:\Users\aishw\entri_python\Python Assignments> python Task4.15.py
Original list: [19, 'red', 12, 'green', 'blue', 10, 'white', 'green', 1]
Result: [1, 10, 12, 19, 'blue', 'green', 'green', 'red', 'white']
PS C:\Users\aishw\entri_python\Python Assignments> 
```

## Exercise 16:

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}

```
Task4.16.py > ...
1  def count_occurrences(nums):
2      result = dict(map(lambda el : (el, list(nums).count(el)), nums))
3      return result
4  nums = [3,4,5,8,0,3,8,5,0,3,1,5,2,3,4,2]
5  print("Original list:",nums)
6  print("Occurrences: ",count_occurrences(nums))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER powershell + \

```
PS C:\Users\aisw\entri_python\Python Assignments> python Task4.16.py
Original list: [3, 4, 5, 8, 0, 3, 8, 5, 0, 3, 1, 5, 2, 3, 4, 2]
Occurrences: {3: 4, 4: 2, 5: 3, 8: 2, 0: 2, 1: 1, 2: 2}
PS C:\Users\aisw\entri_python\Python Assignments> █
```

## Exercise 17:

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]

```
Task4.17.py > lists
1  def remove_none(lists):
2      result = filter(lambda v: v is not None, lists)
3      return list(result)
4
5  lists = [12, 0, None, 23, None, -55, 234, 89, None, 0, 6, -12]
6  print("Original list:", lists)
7  print("Result: ")
8  print(remove_none(lists))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER powershell

```
PS C:\Users\aisw\entri_python\Python Assignments> python Task4.17.py
Original list: [12, 0, None, 23, None, -55, 234, 89, None, 0, 6, -12]
Result:
[12, 0, 23, -55, 234, 89, 0, 6, -12]
PS C:\Users\aisw\entri_python\Python Assignments>
```