Assignment – 7

Program -1

```
#1. Write a python function that returns the index of
 the smallest element in a list of
#integers. If the number of such elements is greater than 1,
return the smallest index.
#Use the following function header:
#def indexOfSmallestElement(lst):

# Iterate through lst using for loop
# Find the smallest element and assign it to min_no
# Find if there are any other elements equal to min_no, n ==
min_no
# append index of n to index_lst
# Sort the index_lst
# return the smallest element from the index_lst


def indexOfSmallestElement(lst):
    org_lst = lst
    index_lst = list()
    # sorted list
    sorted_lst = sorted(org_lst)
    min_no = sorted_lst[0]
    for n in org_lst:
        if n == min_no:
            index_n = org_lst.index(n)
            index_lst.append(index_n)

    sorted_index = sorted(index_lst)
    return sorted_index[0]
```

```
lst_small_ele = [43, 67, 45, -4, 23, 345, 0, 1132, 45322, 6, 2,
3, 2, 1, 0, -4, 4, -4]
index_num = indexOfSmallestElement(lst_small_ele)
print("Smallest Index:",index_num)
```

Output

Smallest Index: 3

Program -2

```
#2. Write the python function mostCommonName, that takes a list
 of names (such as ["Ja
#ne", "Aaron", "Cindy", "Aaron"], and returns the most common
name in this list
#(in this case, "Aaron"). If there is more than one such name,
return a set of the most
#common names. So mostCommonName(["Jane", "Aaron", "Jane",
"Cindy", "Aaron"])
#returns the set {"Aaron", "Jane"}. If the set is empty, return
None. Also, treat names
#case sensitive, so "Jane" and "JANE" are different names.

def mostCommonName(nameLst):
    name_set = set()
    for name in nameLst:
        count = 0
        for comm_name in nameLst:

            if name == comm_name:
                count += 1
                if count > 1:
                    name_set.add(comm_name)

    return name_set


nameLst = ["Jane", "Aaron", "Cindy", "Sunita", "Aaron",
"Sunita"]
common_names =set()
common_names = mostCommonName(nameLst)
print(common_names)
```

Output

{'Sunita', 'Aaron'}

Program -3

```
# 3. Write the python function isPalindromicList(a) that takes
a list and returns True if it is
# the same forwards as backwards and False otherwise.

def isPalindromicList(Lst):

    if Lst == Lst[::-1]:
        return True
    else:
        return False

Lst1 = [1, 3, 4, 5, 5, 4, 3, 1]
Lst2 = [12, 43, 65, 87, 23, 34]
print(Lst1,"is Palindrome List?",isPalindromicList(Lst1))
print(Lst2,"is Palindrome List?",isPalindromicList(Lst2))
```

Output

[1, 3, 4, 5, 5, 4, 3, 1] is Palindrome List? True

[12, 43, 65, 87, 23, 34] is Palindrome List? False