Program 3: write a python Program to find duplicate values from a list and display those

#Define a function

def find_duplicates():

   #using list comprehension

   numbers = [int(x) for x in input("Enter numbers separated by space: ").split()]

   duplicates = [num for num in set(numbers) if numbers.count(num) > 1]

   if duplicates:

     print(f"List is {numbers} and the duplicates are {duplicates}")

   else:

     print("No duplicates found.")

#Calling the function

find_duplicates()

Output:

Enter numbers separated by space: 1 1 2 3 4 4 5 1
List is [1, 1, 2, 3, 4, 4, 5, 1] and the duplicates are [1, 4]

Program 4: write a python Program split a given list into two parts where the length of the first part of the list is given.

Original list:

[1, 1, 2, 3, 4, 4, 5, 1]

Length of the first part of the list: 3

Spitted the said list into two parts:

([1, 1, 2], [3, 4, 4, 5, 1])

```
#Define a function

def split_list(original_list, length):

    return ([x for i, x in enumerate(original_list) if i < length],

        [x for i, x in enumerate(original_list) if i >= length])


original_list = [1, 1, 2, 3, 4, 4, 5, 1]

length = 3

part1, part2 = split_list(original_list, length)

print("Original List:", original_list)

print("Length of the first part:", length)

print("Split List:", (part1, part2))
```

Output:

```
Original List: [1, 1, 2, 3, 4, 4, 5, 1]
Length of the first part: 3
Split List: ([1, 1, 2], [3, 4, 4, 5, 1])
```

Program 4: write a python Program to traverse a given list in reverse order, and print the element with the original index.

Original list:

['red', 'green', 'white', 'black']

Traverse the said list in reverse order:

black
white
green
red
```
def traverse_reverse(lst):

    for i, elem in enumerate(reversed(lst)):

        print(f"Index {len(lst) - i - 1}: {elem}")


original_list = ['red', 'green', 'white', 'black']

print("Original List:", original_list)

print("Traverse in Reverse Order:")

traverse_reverse(original_list)
```

Output:

```
Original List: ['red', 'green', 'white', 'black']
Traverse in Reverse Order:
Index 3: black
Index 2: white
Index 1: green
Index 0: red
```