In [1]:
```
def is_Sublist(list_a, sublist_b):
    if len(sublist_b) > len(list_a):
        return False

    for i in range(len(list_a) - len(sublist_b) + 1):
        if sublist_b == list_a[i:i+len(sublist_b)]:
            return True

    return False

a = [2, 4, 3, 5, 7]
b = [4, 3]
c = [3, 7]

print(is_Sublist(a, b))
print(is_Sublist(a, c))
```

```
True
False
```

In [2]:
```
def find_common_items(list1, list2):
    set1 = set(list1)
    set2 = set(list2)
    common_items = set1.intersection(set2)
    return common_items

color1 = ["Red", "Green", "Orange", "White"]
color2 = ["Black", "Green", "White", "Pink"]

common_colors = find_common_items(color1, color2)
print(common_colors)
```

```
{'Green', 'White'}
```

In [3]:
```
def find_list_difference(list1, list2):
    set1 = set(list1)
    set2 = set(list2)
    difference = set1 - set2
    return list(difference)

list1 = [1, 2, 3, 4]
list2 = [1, 2]

difference = find_list_difference(list1, list2)
print(difference)
```

```
[3, 4]
```

In [4]:
```
def generate_permutations(l):
    if len(l) == 0:
        return [[]]

    permutations = []
    for i in range(len(l)):
        remaining = l[:i] + l[i+1:]
        for j in generate_permutations(remaining):
            permutations.append([l[i]] + j)

    return permutations

input_list = [1, 2, 3]
permutations_list = generate_permutations(input_list)
print(permutations_list)
```

```
[[1, 2, 3], [1, 3, 2], [2, 1, 3], [2, 3, 1], [3, 1, 2], [3, 2, 1]]
```

In [5]:
```
def remove_duplicates(lst):
    return list(set(lst))

a = [10, 20, 30, 20, 10, 50, 60, 40, 80, 50, 40]
result = remove_duplicates(a)
print(result)
```

```
[40, 10, 80, 50, 20, 60, 30]
```