In [17]:

```
def is_sublist(main_list, sublist):
    if not sublist:
        return True
    if not main_list:
        return False
    if sublist == main_list[:len(sublist)]:
        return True
    return is_sublist(main_list[1:], sublist)


main_list = [2,4,3,5,7]
sublist1 = [4,3]
sublist2 = [3,7]

if is_sublist(main_list, sublist1):
    print("Sublist 1 is present in the main list.")
else:
    print("Sublist 1 is not present in the main list.")

if is_sublist(main_list, sublist2):
    print("Sublist 2 is present in the main list.")
else:
    print("Sublist 2 is not present in the main list.")
```

```
Sublist 1 is present in the main list.
Sublist 2 is not present in the main list.
```

In [18]:

```
def find_common_items(list1, list2):
    common_items = []
    for item in list1:
        if item in list2:
            common_items.append(item)
    return common_items

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

common_colors = find_common_items(color1, color2)
print("Common colors:", common_colors)
```

```
Common colors: ['Green', 'White']
```

In [20]:

```
def difference(list1, list2):
    return list(set(list1) - set(list2))

list1 = [1, 2, 3, 4]
list2 = [1, 2]
subtraction = difference(list1, list2)
print("Difference between list1 and list2:", subtraction)
```

```
Difference between list1 and list2: [3, 4]
```

In [25]:

```
from itertools import permutations

def generate_permutations(lst):

    perm = permutations(lst)
    perm_list = list(perm)

    return perm_list

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

for perm in permutations_list:
    print(perm)
```

```
(1, 2, 3)
(1, 3, 2)
(2, 1, 3)
(2, 3, 1)
(3, 1, 2)
(3, 2, 1)
```

In [26]:

```
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]
```