

1. Check if the list is already sorted or not. Ascending or descending

```
def check_sorted(lst):  
    if lst == sorted(lst):  
        return "Ascending"  
    elif lst == sorted(lst, reverse=True):  
        return "Descending"  
    else:  
        return "Not sorted"  
print(check_sorted([1, 2, 3, 4]))  
print(check_sorted([9, 8, 7, 6]))  
print(check_sorted([1, 3, 2, 4]))
```

2. Missing number in a list. [1, 2, 3, 5, 6, 7, 9, 8]

1. Sum of n numbers using formulae - Sum of elements in list

```
def find_missing_number(lst):  
    n = len(lst) + 1  
    total_sum = n * (n + 1) // 2  
    list_sum = sum(lst)  
    return total_sum - list_sum  
print(find_missing_number([1, 2, 3, 5, 6, 7, 9, 8]))
```

2. 2 loops method

```
def find_missing_number(lst):  
    for num in range(1, len(lst) + 2):  
        if num not in lst:  
            return num  
print(find_missing_number([1, 2, 3, 5, 6, 7, 9, 8]))
```

3. xor method.

```
def find_missing_number(lst):
```

```

n = len(lst) + 1
xor_all = 0
xor_list = 0

for num in range(1, n + 1):
    xor_all ^= num
for num in lst:
    xor_list ^= num
return xor_all ^ xor_list
print(find_missing_number([1, 2, 3, 5, 6, 7, 9, 8]))

```

#### 4. Sorting

```

def find_missing_number(lst):
    lst.sort()
    for i in range(len(lst) - 1):
        if lst[i] + 1 != lst[i + 1]:
            return lst[i] + 1
print(find_missing_number([1, 2, 3, 5, 6, 7, 9, 8]))

```

#### 3. Check if an array is a subset of another or not.

```

def is_subset(list1, list2):
    return set(list1).issubset(set(list2))
print(is_subset([1, 2, 3], [1, 2, 3, 4, 5]))
print(is_subset([1, 6], [1, 2, 3, 4, 5]))

```

#### 4. Check if a + b = target exists in a list

```

def check_sum_exists(lst, target):
    seen = set()
    for num in lst:
        if target - num in seen:
            return True

```

```
    seen.add(num)

    return False

print(check_sum_exists([1, 2, 3, 4, 5], 9))
print(check_sum_exists([1, 2, 3, 4, 5], 10))
```