task-1

In [3]:
```
def count_numbers(numbers):
    count_even = []
    count_odd = []
    count_prime = []

    for num in numbers:
        if num % 2 == 0:
            count_even.append(num)
        else:
            count_odd.append(num)

        if num > 1:
            is_prime = True
            for i in range(2, int(num**0.5) + 1):
                if num % i == 0:
                    is_prime = False
                    break
            if is_prime:
                count_prime.append(num)

    return count_even, count_odd, count_prime


# Example usage
num_list = [2, 5, 8, 11, 12, 17, 20]
even_count, odd_count, prime_count = count_numbers(num_list)

print("Even numbers:", even_count)
print("Odd numbers:", odd_count)
print("Prime numbers:", prime_count)
```

```
Even numbers: [2, 8, 12, 20]
Odd numbers: [5, 11, 17]
Prime numbers: [2, 5, 11, 17]
```

task-2

In [5]:
```
listt = ["Jamal", "Hamza","Ali","c", "P"]
name= []
char = []
for i in listt:
    if i in "Jamal, Hamza, Ali":
        name.append(i)
    elif i in "c,P":
        char.append(i)
print(name,char)
```

```
['Jamal', 'Hamza', 'Ali'] ['c', 'P']
```

task-3

In [17]:
```
List1= [5,9,8,6,0,2,2,'a','b','c','Jamal']
List2 = []
for i in List1:
    data_type = type(i).__name__
    List2.append(data_type)
List2
```

Out[17]:
```
['int', 'int', 'int', 'int', 'int', 'int', 'int', 'str', 'str', 'str', 'str']
```

task-4

In [18]:
```
def analyze_list(lst):
    # Most repeating values
    count = {}
    most_repeating = []

    for value in lst:
        if value in count:
            count[value] += 1
        else:
            count[value] = 1

    max_count = max(count.values())
    most_repeating = [value for value, freq in count.items() if freq == max_cou

    # NaN values
    nan_values = [value for value in lst if isinstance(value, float) and value

    # Highest value
    highest_value = max(lst)

    # Prime numbers
    def is_prime(n):
        if n < 2:
            return False
        for i in range(2, int(n ** 0.5) + 1):
            if n % i == 0:
                return False
        return True

    prime_numbers = [value for value in lst if isinstance(value, int) and is_pr

    return most_repeating, nan_values, highest_value, prime_numbers


# Example usage
List = [1, float("nan"), 4, 5, 6, 0, 6, 7]
most_repeating_values, nan_values, highest_value, prime_numbers = analyze_list(

print("List:", List)
print("Most Repeating Values:", most_repeating_values)
print("NaN Values:", nan_values)
print("Highest Value:", highest_value)
print("Prime Numbers:", prime_numbers)
```

```
List: [1, nan, 4, 5, 6, 0, 6, 7]
Most Repeating Values: [6]
NaN Values: [nan]
Highest Value: 7
Prime Numbers: [5, 7]
```

task-5

In [20]:
```
def analyze_list(lst):
    numeric_values = [value for value in lst if isinstance(value, (int, float))
    character_values = [value for value in lst if isinstance(value, str) and le
    string_values = [value for value in lst if isinstance(value, str)]

    return numeric_values, character_values, string_values


# Example usage
List = [8, 10, 'a', 'b', 's', 'f', 99, 'fine', 'Education', 6, 53, 'Pass']
numeric_values, character_values, string_values = analyze_list(List)

print("List:", List)
print("Numeric Values:", numeric_values)
print("Character Values:", character_values)
print("String Values:", string_values)
```

```
List: [8, 10, 'a', 'b', 's', 'f', 99, 'fine', 'Education', 6, 53, 'Pass']
Numeric Values: [8, 10, 99, 6, 53]
Character Values: ['a', 'b', 's', 'f']
String Values: ['a', 'b', 's', 'f', 'fine', 'Education', 'Pass']
```

In [ ]: