```
 Task 1:
        Write a function which take numerical list and return number of list even odd prime
        Task 2:
        Write a python function which take this list = ["Jamal", "Hamza","Ali","c", "P"] and return
two list one
        for string and other of characters like String_list=["Jamal","Hamza","Ali"] and character_li
st =["c","P"]
        Task 3:
        Write a function which return list each element datatype in new list same at that position o
f element.
        List1= [5,9,8,6,0,2,2,a,b,c,Jamal] so according to this solution will be this
        Hint:list1_solution=[int,int,int,int,int,int,int,ch,ch,ch,str]
        Task 4:
        Make a list, List= [ 1,nan,4,5,6,0,6,7]
        The function should return:
        a)Most Repeating values
        b)Display nan values
        c)Highest value
        d)Prime number
        Task 5:
        [8,10,a,b,s,f,99,'fine','Education',6,53,'Pass']
        a)The Function should return only
        b)Numeric Value
        c)Character
        d)string
```

Task 1:

```
 Write a function which take numerical list and return number of list even odd prime
```

In [1]:
```
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

def find_even_odd_prime(numbers):
    even_numbers = []
    odd_numbers = []
    prime_numbers = []

    for num in numbers:
        if num % 2 == 0:
            even_numbers.append(num)
        else:
            odd_numbers.append(num)

        if is_prime(num):
            prime_numbers.append(num)

    return even_numbers, odd_numbers, prime_numbers

numbers = [1, 3, 5, 45, 6, 4, 23]
even_nums, odd_nums, prime_nums = find_even_odd_prime(numbers)

print("Even numbers:", even_nums)
print("Odd numbers:", odd_nums)
print("Prime numbers:", prime_nums)
```

```
Even numbers: [6, 4]
Odd numbers: [1, 3, 5, 45, 23]
Prime numbers: [3, 5, 23]
```

Task 2:

Write a python function which take this list = ["Jamal", "Hamza","Ali","c", "P"] and return two list one
for string and other of characters like String_list=["Jamal","Hamza","Ali"] and character_list = ["c","P"]

In [15]:
```
def separate_list(lst):
    string_list = []
    character_list = []

    for item in lst:
        if len(item) > 1:
            string_list.append(item)
        else:
            character_list.append(item)

    return string_list, character_list

string_list, character_list = separate_list(["Jamal", "Hamza", "Ali", "c", "P"])
print("String List:", string_list)
print("Character List:", character_list)
```

```
String List: ['Jamal', 'Hamza', 'Ali']
Character List: ['c', 'P']
```

Task 3:

Write a function which return list each element datatype in new list same at that position of element.
List1= [5,9,8,6,0,2,2,a,b,c,Jamal] so according to this solution will be this
Hint:list1_solution=[int,int,int,int,int,int,int,ch,ch,ch,str]

In [32]:
```
def get_datatypes(lst):
    result = []

    for item in lst:
        datatype = type(item).__name__
        if datatype == "int":
            result.append('int')
        elif datatype == "str" and len(item) > 1:
            result.append('str')
        else:
            result.append('ch')

    return result

lst = [5, 9, 8, 6, 0, 2, 2, 'a', 'b', 'c', 'Jamal']
solution = get_datatypes(lst)
print(solution)
```

```
['int', 'int', 'int', 'int', 'int', 'int', 'int', 'ch', 'ch', 'ch', 'str']
```

Task 4:

Make a list, List= [1,nan,4,5,6,0,6,7]
The function should return:
a)Most Repeating values
b)Display nan values
c)Highest value
d)Prime number

In [3]:
```
def analyze_list(lst):
    cleaned_list = []
    for item in lst:
        if item is not None:
            cleaned_list.append(item)

    counts = {}
    for item in cleaned_list:
        if item in counts:
            counts[item] += 1
        else:
            counts[item] = 1



    max_count = 0
    most_repeating = []
    for item, count in counts.items():
        if count > max_count:
            max_count = count
            most_repeating = [item]
        elif count == max_count:
            most_repeating.append(item)



    nan_values = []
    for item in lst:
        if item is None:
            nan_values.append(item)



    highest_value = 0
    for item in cleaned_list:
        if item > highest_value:
            highest_value = item


    prime_numbers = []
    for item in cleaned_list:
        if item < 2:
            continue
```

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

    return most_repeating, nan_values, highest_value, prime_numbers

lst = [1, None, 4, 5, 6, 0, 6, 7]
results = analyze_list(lst)
print("Most Repeating Values:", results[0])
print("NaN Values:", results[1])
print("Highest Value:", results[2])
print("Prime Numbers:", results[3])
```

```
Most Repeating Values: [6]
NaN Values: [None]
Highest Value: 7
Prime Numbers: [5, 7]
```

Task 5:

```
[8,10,a,b,s,f,99,'fine','Education',6,53,'Pass']
a)The Function should return only
b)Numeric Value
c)Character
d)string
```

In [8]:
```
def analyze_list(lst):
    only_values = []
    numeric_values = []
    character_values = []
    string_values = []

    for item in lst:

        if isinstance(item, (int, str)):
            only_values.append(item)


        if isinstance(item, int):
            numeric_values.append(item)


        if isinstance(item, str) and len(item) == 1:
            character_values.append(item)


        if isinstance(item, str) and len(item) > 1:
            string_values.append(item)

    return only_values, numeric_values, character_values, string_values

my_list = [8, 10, 'a', 'b', 's', 'f', 99, 'fine', 'Education', 6, 53, 'Pass']
result = analyze_list(my_list)
print("Only Values:", result[0])
print("Numeric Values:", result[1])
print("Character Values:", result[2])
print("String Values:", result[3])
```

```
Only Values: [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: ['fine', 'Education', 'Pass']
```

In [ ]: