1. Write a python Function to list even and odd numbers in a list.

In [6]:

```
even=[]
odd=[]
def evod(list):
    for i in list:
        if(i%2==0):
            even.append(i)
        else:
            odd.append(i)
list=[1,2,3,4]
evod(list)

print(even)
print(odd)
```

```
[2, 4]
[1, 3]
```

2. Write and run a Python program that asks the user to enter 8 integers (one at a time), and then prints out how many of those integers were even numbers. For example, if the user entered 19, 6, 9, 20, 13, 7, 6, and 1, then your program should print out 3 since 3 of those numbers were even.

In [16]:

```
print('Enter 8 Integers')
array=[]
count=0;
for i in range (0,8):
    x=int(input())
    array.append(x)
    if(x%2==0):
        count+=1
count
```

```
Enter 8 Integers
4
2
6
8
5
7
3
9
```

Out[16]:

```
4
```

3. Write a Python program where you take any positive integer n, if n is even, divide it by 2 to get n / 2. If n is odd, multiply it by 3 and add 1 to obtain 3n + 1. Repeat the process until you reach 1.

In [21]:

```
def fun(n):
    while(n>1):
        if(n%2==0):
            n=n/2
        else:
            n=(n*3)+1
        fun(n)

print('Enter positive integer')
x=int(input())
fun(x)
```

```
Enter positive integer
8
```

4. Write a Python program to compute the sum of all the multiples of 3 or 5 below 500.

In [23]:

```
sum=0;
for i in  range(500):
    if(i%15==0):
        sum+=i
sum
```

Out[23]:

```
8415
```

5. To write a Python program to find first 'n' prime numbers from a list of given numbers.

In [30]:

```
list=[2,4,5,6,7,8,9,11,13,12]
n=3
n1=0
for i in list:
    j=2;
    count=0;
    while(j<=i):
        if(i%j==0):
            count+=1
        j+=1
    if(count==1 and n1 < n ):
        print(i)
        n1=n1+1
```

```
2
5
7
```

6. To write a Python program to compute matrix multiplication.

In [8]:

```
def matrix_mul(matrix1, matrix2):
    if len(matrix1[0]) != len(matrix2):
        raise ValueError("check matrix size")

    result = [[0 for k in range(len(matrix2[0]))] for j in range(len(matrix1))]

    for i in range(len(matrix1)):
        for j in range(len(matrix2[0])):
            for k in range(len(matrix2)):
                result[i][j] += matrix1[i][k] * matrix2[k][j]

    return result


matrix_A = [[1, 2, 3],
            [4, 5, 6]]

matrix_B = [[7, 8],
            [9, 10],
            [11, 12]]

result_matrix = matrix_mul(matrix_A, matrix_B)

for row in result_matrix:
    print(row)
```

```
[58, 64]
[139, 154]
```

7. Write a python Function to count the number of vowels in a string.

In [5]:

```
def fun(str):
    count=0
    for x in str:
        if(x == 'a' or x == 'e' or x == 'i' or x == 'o' or x == 'u' or x == 'A' or x =
            count+=1
    print(count)



fun('Mphasis')
```

```
2
```

8. Write a python Function for finding factorial for the given number using a recursive function.

In [50]:

```
def fact(n):
    if(n>1):
        return (n*fact(n-1))
    else :
        return 1

print('Enter the number')
x=int(input())
result=fact(x)
result
```

```
Enter the number
4
```

Out[50]:

```
24
```

9. Write a python Function for generating the Fibonacci series using the function.

In [89]:

```
def fibo(n):
    a,b = 0,1
    print(a)
    print(b)
    while b <= n:
        a,b=b,a+b
        if(b<=n):
            print(b)

fibo(5)
```

```
0
1
1
2
3
5
```

10. Python program to display the given integer in reverse order using the function without an in- builtfunction.

In [72]:

```
def reverseno(n):
    rev=0;
    while n!=0:
        digit=n%10
        rev=rev*10+digit
        n=n//10
    print(rev)

reverseno(1234)
```

```
4321
```

11. Write a Python Function to display all integers within the range 200-300 whose sum of digits is an even number.

In [107…

```
def fun():
    for i in range(200,300):
        sum=0
        sum=sum+(i%10)
        sum=sum+((i//10)%10)
        sum=sum+(i//100)
        if(sum%2 == 0):
            print(i)

fun()
```

```
200
202
204
206
208
211
213
215
217
219
220
222
224
226
228
231
233
235
237
239
240
242
244
246
248
251
253
255
257
259
260
262
264
266
268
271
273
275
277
279
280
282
284
286
288
291
293
295
297
299
```

12. Write a python Function to find the number of digits and sum of digits for a given integer.

In [10]:

```
def fun(n):
    sum=0
    count=0
    while n != 0:
        c=n%10
        sum+=c
        n=n//10
        count+=1
    print('sum of digits is :',sum)
    print('No of digits is:',count)

fun(314)
```

```
sum of digits is : 8
No of digits is: 3
```

13. Write functions called is*sorted that takes a list as a parameter and returns True if the list is sorted in ascending order and False otherwise and has* duplicates that takes a list and returns True if there is any element that appears more than once. It should not modify the original list.

In [64]:

```
import copy

def is_sorted(lst):
    list1=copy.deepcopy(lst)
    list1.sort()
    print(lst)
    print(list1)
    if (lst==list1):
        print('TRUE')
    else:
        print('FALSE')

    for i in list1:
        count=0
        for j in list1:
            if(i == j):
                count+=1
        if count > 1:
            flag=1
    if flag:
        print('TRUE')



lst=[1,5,6]
is_sorted(lst)
```

```
[1, 5, 6]
[1, 5, 6]
TRUE
```

```
---------------------------------------------------------------------------
UnboundLocalError                         Traceback (most recent call last)
Cell In[64], line 26
     21         print('TRUE')
     25 lst=[1,5,6]
---> 26 is_sorted(lst)

Cell In[64], line 20, in is_sorted(lst)
     18     if count > 1:
     19         flag=1
---> 20 if flag:
     21     print('TRUE')

UnboundLocalError: cannot access local variable 'flag' where it is not associated wit
h a value
```

14. Write functions called nested_sum that takes a list of integers and adds up the elements from all the nested lists and cumsum that takes a list of numbers and returns the cumulative sum; that is, a new list where the ith element is the sum of the first i + 1 elements from the original list.

In [74]:

```python
import sys

def nested_sum(lst):
    total = 0
    for item in lst:
        if isinstance(item,type(lst)):
            total += nested_sum(item)
        else:
            total += item
    return total

def cumsum(lst):
    cumulative_sum = []
    total = 0
    for num in lst:
        total += num
        cumulative_sum.append(total)
    return cumulative_sum


nested_list = [1, [2, 3], [4, [5, 6]], 7]
cumulative_list = [1, 2, 3, 4, 5]

result_nested_sum = nested_sum(nested_list)
result_cumsum = cumsum(cumulative_list)
```

```
print("Nested Sum:", result_nested_sum)
print("Cumulative Sum:", result_cumsum)
```

```
Nested Sum: 28
Cumulative Sum: [1, 3, 6, 10, 15]
```

In [ ]: