1.Write a python unction to list even and odd numbers from a list

```python
def print_even_odd(numbers):
    even_numbers = []
    odd_numbers = []

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

    print("Even numbers:", even_numbers)
    print("Odd numbers:", odd_numbers)

# Example list of numbers
my_list = [1, 2, 3, 4, 5, 6, 7, 8, 9]

# Call the function with the list
print_even_odd(my_list)
```

```
Even numbers: [2, 4, 6, 8]
Odd numbers: [1, 3, 5, 7, 9]
```

2.Python program to print even numbers from 8 input numbers by the user

```python
# Function to check if a number is even
def is_even(number):
    return number % 2 == 0

# Function to get 8 integers from the user
def get_integers():
    integers = []
    for i in range(8):
        try:
            integer = int(input(f"Enter integer {i + 1}: "))
            integers.append(integer)
        except ValueError:
            print("Invalid input. Please enter an integer.")
            i -= 1 # Try again for the same index
    return integers

# Main function
def main():
    integers = get_integers()
    even_count = sum(1 for num in integers if is_even(num))
    print(f"Out of the 8 integers you entered, {even_count} of them are even numbers.")

if __name__ == "__main__":
    main()
```

```
Enter integer 1: 11
Enter integer 2: 22
Enter integer 3: 33
Enter integer 4: 44
Enter integer 5: 55
Enter integer 6: 66
Enter integer 7: 77
Enter integer 8: 8
Out of the 8 integers you entered, 4 of them are even numbers.
```

3.python program to print 1 at last

```python
def num(n):
    while n != 1:
        print(n, end=' ')
        if n % 2 == 0:
            n = n // 2
        else:
            n = 3 * n + 1
    print(1) # Print the final '1'

try:
    n = int(input("Enter a positive integer: "))
    if n <= 0:
        print("Please enter a positive integer.")
    else:
        print("Collatz sequence:")
```

```
        num(n)
except ValueError:
    print("Invalid input. Please enter a positive integer.")
```

```
Enter a positive integer: 5
Collatz sequence:
5 16 8 4 2 1
```

4.write a python program to compute sum of all multiples of 3 and 5 below 500

In [8]:
```python
def is_multiple(number):
    return number % 3 == 0 or number % 5 == 0
def sum_multiples(limit):
    total_sum = 0
    for i in range(1, limit):
        if is_multiple(i):
            total_sum += i
    return total_sum
if __name__ == "__main__":
    limit = 500
    result = sum_multiples(limit)
    print(f"The sum of all multiples of 3 and 5 under {limit} is:", result)
```

```
The sum of all multiples of 3 and 5 under 500 is: 57918
```

5.write a python program to print first n prime numbers from the given list

In [83]:
```python
def is_prime(num):
    if num <= 1:
        return False
    for i in range(2, int(num**0.5) + 1):
        if num % i == 0:
            return False
    return True

def find_primes_in_list(numbers, n):
    prime_numbers = []
    for num in numbers:
        if is_prime(num):
            prime_numbers.append(num)
            if len(prime_numbers) == n:
                break
    return prime_numbers

numbers = [1,2,3,4,5,6,7]


n = 4

prime_list = find_primes_in_list(numbers, n)
print(f"The first {n} prime numbers from the list are: {prime_list}")
```

```
The first 4 prime numbers from the list are: [2, 3, 5, 7]
```

6.write a python program to compute matrix multiplication

In [76]:
```python
def matrix_multiply(A, B):

    result = [[0 for j in range(len(B[0]))] for i in range(len(A))]


    def multiply(A, B, result, i, j, k):
        if i >= len(A):
            return
        if j >= len(B[0]):
            return multiply(A, B, result, i+1, 0, 0)
        if k >= len(B):
            return multiply(A, B, result, i, j+1, 0)
        result[i][j] += A[i][k] * B[k][j]
        multiply(A, B, result, i, j, k+1)


    multiply(A, B, result, 0, 0, 0)
    return result


A = [[12, 7, 3], [4, 5, 6], [7, 8, 9]]
B = [[5, 8, 1, 2], [6, 7, 3, 0], [4, 5, 9, 1]]
```

```
result = matrix_multiply(A, B)
for row in result:
    print(row)
```

```
[114, 160, 60, 27]
[74, 97, 73, 14]
[119, 157, 112, 23]
```

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

In [25]:
```python
import re
def Check_Vow(string, vowels):
    str_list = re.findall(f'[{vowels}]', string, re.I)
    print(len(str_list))
    return str_list
vowels = 'aeiou'
string = "Shaik Tauheer Ahamed"
print (Check_Vow(string, vowels))
```

```
9
['a', 'i', 'a', 'u', 'e', 'e', 'A', 'a', 'e']
```

8.Write a python function to find the factorial of a number

In [29]:
```python
def fact(n):
    if n<=1:
        return n
    else:
        return n*fact(n-1)
print("The factorial of is "  )
fact(4)
```

```
The factorial of is
```
Out[29]: 24

9. write python function to generate fibanocci series of a number

In [38]:
```python
def fibo_s(n):
    if n <= 1:
        return n
    else:
        return(fibo_s(n-1) + fibo_s(n-2))
nterms = 5

if nterms <= 0:
    print("Plese enter a positive integer")
else:
    print("Fibonacci sequence:")
    for i in range(nterms):
        print(fibo_s(i))
```

```
Fibonacci sequence:
0
1
1
2
3
```

10.Python program to print a number in reverse with using user defined function and without using inbuilt function

In [41]:
```python
reverse, base = 0, 1
def findReverse(num):
    global reverse
    global base
    if(num > 0):
        findReverse((int)(num/10))
        reverse += (num % 10) * base
        base *= 10
    return reverse
num = int(input('Enter a number: '))
print('The reverse number is =', findReverse(num))
```

```
Enter a number: 456
The reverse number is = 654
```

11.Write a python program to print numbers between 200 to 300

In [22]:
```python
def check(n):
    sum = 0

    while n!=0:
        sum+=n%10
        n=n//10

    if sum%2==0:
        return True

    else:
        return False

def solve():
    values = []
    for i in range(200, 301):
        if check(i):
            values.append(i)
    return values


ans = solve()
print("The value whose sum is even in the range 200 - 300")
print(ans)
```

```
The value whose sum is even in the range 200 - 300
[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, 2
88, 291, 293, 295, 297, 299]
```

12 Write a python program to find number of digits and sum of digits for a given integer

In [66]:
```python
def getSum(n):
    count=0
    sum = 0
    for digit in str(n):
        sum += int(digit)
        count=count+1
    print("The sum of digits is ",sum)
    print("The number of digits are",count)

n = 12345
print(getSum(n))
```

```
The sum of digits is  15
The number of digits are 5
None
```

In [ ]:
```
13.
```

In [15]:
```python
def is_sorted(nums):

    for i in range(1, len(nums)):

        if (nums[i]<nums[i-1]):
            return False

    return True

def has_duplicates (nums):

    vis = set()

    for i in nums:
        if i in vis:

            return True


        vis.add(i)
    return False
##################################################

n = int(input("Enter the size of the list: "))

nums1=[]

for i in range(0, n):
```

```python
        ele = int(input("Enter the number in to the list1: "))

        nums1.append(ele)

    #calling is sorted function

    ans1= is_sorted(nums1)

    if ans1:

        print("\nUsing is sorted fucntion the list\n (nums1}\n is a sorted list")

    else:
        print("\nUsing is sorted fucntion the list in (nums1)\n is not a sorted list")


    ###############################################################################
    nums2=[]

    for i in range(0, n):

        ele = int(input("Enter the number in to the list2: "))

        nums2.append(ele)

    #calling has duplicates function

    ans2= has_duplicates(nums2)

    if ans2:

        print(f"using has_duplicates fucntion the list\n {nums2} in has duplicates in it")

    else:
        print(f"using has_duplicates fucntion the list\n {nums2}in has no duplicates in it")
```

```
Enter the size of the list: 11
Enter the number in to the list1: 15
Enter the number in to the list1: 52
Enter the number in to the list1: 55
Enter the number in to the list1: 52
Enter the number in to the list1: 52
Enter the number in to the list1: 56
Enter the number in to the list1: 86
Enter the number in to the list1: 56
Enter the number in to the list1: 78
Enter the number in to the list1: 85
Enter the number in to the list1: 55

Using is sorted fucntion the list in (nums1}
 is not a sorted list
Enter the number in to the list2: 54
Enter the number in to the list2: 85
Enter the number in to the list2: 85
Enter the number in to the list2: 56
Enter the number in to the list2: 56
Enter the number in to the list2: 52
Enter the number in to the list2: 85
Enter the number in to the list2: 55
Enter the number in to the list2: 86
Enter the number in to the list2: 44
Enter the number in to the list2: 78
using has_duplicates fucntion the list
 [54, 85, 85, 56, 56, 52, 85, 55, 86, 44, 78] in has duplicates in it
```

14

In [20]:
```python
def nested_sum(nums):

    sum = 0

    for i in nums:
            if isinstance(i, list):

                    sum +=nested_sum(i)

            elif isinstance(i, int):

                    sum += i

    return sum

# creating nested list for the fun nested sum
nums1 = [1, [2, 3], [3, 4, 5], [4, 5, 6, 7]]

#calling the nested sum function
ans1 = nested_sum(nums1)
```

```python
print(f"Using nested sum then {nums1}\nsum is {ans1}")

#cummsum function

def cum_sum(nums):
    sum = 0

    ans = []

    for i in nums:

        sum+=i

        ans.append(sum)

    return ans

#List for cum Sum
nums2 = [10, 20, 30, 40, 50]

#calling the cum sum function
ans2 = cum_sum(nums2)

print(f"\nUsing cum sum then {nums2}\nanswer list is {ans2}")
```

```
Using nested sum then [1, [2, 3], [3, 4, 5], [4, 5, 6, 7]]
sum is 40

Using cum sum then [10, 20, 30, 40, 50]
answer list is [10, 30, 60, 100, 150]
```

In [ ]: