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

In [5]:

```
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

In [1]:

```
# 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

In [7]:

```
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]:

```
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]:

```
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]:

```
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]:

```
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]:

```
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]:

```
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]:

```
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

11.Write a python program to print numbers between 200 to 300 whose sum is even

In [88]:

```
def sum_of_digits_is_even(num):
    digit_sum = sum(int(digit) for digit in str(num))
    return digit_sum % 2 == 0

def find_numbers_with_even_digit_sum(start,end):
    result = []
    for number in range(start, end + 1):
        if sum_of_digits_is_even(number):
            result.append(number)
    return result

# Define the range
start_range = 200
end_range = 300

# Find and display numbers with even digit sum
even_digit_sum_numbers = find_numbers_with_even_digit_sum(start_range, end_range)
print("Numbers within the range 200 to 300 whose sum of digits is even:")
print(even_digit_sum_numbers)
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[88], line 17
     14 end_range = 300
     16 # Find and display numbers with even digit sum
---> 17 even_digit_sum_numbers = find_numbers_with_even_digit_sum(start_range, end_range)
     18 print("Numbers within the range 200 to 300 whose sum of digits is even:")
     19 print(even_digit_sum_numbers)

Cell In[88], line 8, in find_numbers_with_even_digit_sum(start, end)
      6 result = []
      7 for number in range(start, end + 1):
----> 8     if sum_of_digits_is_even(number):
      9         result.append(number)
     10 return result

Cell In[88], line 2, in sum_of_digits_is_even(num)
      1 def sum_of_digits_is_even(num):
----> 2     digit_sum = sum(int(digit) for digit in str(num))
      3     return digit_sum % 2 == 0

TypeError: sum() missing 1 required positional argument: 'num2'
```

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

In [66]:

```
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.
```

Loading [MathJax]/jax/output/CommonHTML/fonts/TeX/fontdata.js