"""

Problem: Check if a number is prime and filter prime numbers from a list.

"""

import math

def is\_prime(n):

if n < 2:

return False

if n == 2:

return True

if n % 2 == 0:

return False

for i in range(3, int(math.sqrt(n)) + 1, 2):

if n % i == 0:

return False

return True

numbers = [10, 15, 2, 3, 5]

primes = [n for n in numbers if is\_prime(n)]

print(primes)

"""

Problem: Generate the Fibonacci sequence up to 'n' terms.

"""

def fibonacci(n):

a, b = 0, 1

for \_ in range(n):

print(a, end=" ")

a, b = b, a + b

fibonacci(100)

"""

Problem: Find the next 5 prime numbers after a given number.

"""

def is\_prime(n):

if n < 2:

return False

if n == 2:

return True

if n % 2 == 0:

return False

for i in range(3, int(math.sqrt(n)) + 1, 2):

if n % i == 0:

return False

return True

n = int(input("Enter a number: "))

count = 0

num = n + 1

while count < 5:

if is\_prime(num):

print(num, end=" ")

count += 1

num += 1

"""

Problem: Calculate the factorial of a given number.

"""

def factorial(n):

return 1 if (n==1 or n==0) else n \* factorial(n-1)

n = int(input())

print(factorial(n))

"""

Problem: Generate the nth Fibonacci number.

"""

def fibonacci(n):

if n <= 0:

return "Invalid input"

null = 0

value = 1

ans = 0

for \_ in range(n):

ans = null + value

null = value

value = ans

return ans

n = int(input())

print(fibonacci(n))

"""

Problem: Find the maximum element in an array.

"""

def find\_max(arr):

if not arr:

return None

return max(arr)

arr = [10, 20, 5, 30, 15]

print(find\_max(arr))

"""

Problem: Find the minimum element in an array.

"""

arr = [3, 1, 7, 0, -5, 8]

minimum = min(arr)

print(minimum)

"""

Problem: Calculate the sum of all elements in an array.

"""

def array\_sum(arr):

return sum(arr)

arr = [1, 2, 3, 4, 5]

print(array\_sum(arr))

arr = [2,6,2,8,6]

print(array\_sum(arr))

"""

Problem: Reverse an array.

"""

arr = [1, 2, 3, 4, 5]

arr.reverse()

print(arr)

"""

Problem: Print all factors of a number.

"""

def print\_factors(n):

print(f"Factors of {n} are:")

for i in range(1, n + 1):

if n % i == 0:

print(i)

num = int(input())

print\_factors(num)

"""

Problem: Find the greatest common divisor (GCD) of two numbers using an iterative approach.

"""

def gcd\_iterative(a, b):

while b:

a, b = b, a % b

return a

num1 = int(input())

num2 = int(input())

print(f"{gcd\_iterative(num1, num2)}")

"""

Problem: Find the least common multiple (LCM) of an array of numbers.

"""

def find\_lcm(num1, num2):

if num1 > num2:

num = num1

den = num2

else:

num = num2

den = num1

rem = num % den

while rem != 0:

num = den

den = rem

rem = num % den

gcd = den

lcm = int(num1 \* num2 / gcd)

return lcm

l = [2, 7, 3, 9, 4]

num1 = l[0]

num2 = l[1]

lcm = find\_lcm(num1, num2)

for i in range(2, len(l)):

lcm = find\_lcm(lcm, l[i])

print(lcm)

"""

Problem: Check if a number is a power of two.

"""

class Solution:

def isPowerOfTwo(self, n: int) -> bool:

if n <= 0:

return False

return (n & (n - 1)) == 0

"""

Problem: Check if a number is a power of three.

"""

class Solution:

def isPowerOfThree(self, n: int) -> bool:

if n <= 0:

return False

while n % 3 == 0:

n //= 3

return n == 1

"""

Problem: Implement selection sort.

"""

class Solution:

def selectionSort(self, arr):

n = len(arr)

for i in range(n):

min\_index = i

for j in range(i + 1, n):

if arr[j] < arr[min\_index]:

min\_index = j

arr[i], arr[min\_index] = arr[min\_index], arr[i]

return arr

"""

Problem: Implement insertion sort.

"""

def insertion\_sort(arr):

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

key = arr[i]

j = i - 1

while j >= 0 and key < arr[j]:

arr[j + 1] = arr[j]

j -= 1

arr[j + 1] = key