Here are the Python programs for your requests:

**1. Find the Factorial of a Number**

# Function to find factorial

def factorial(num):

if num == 0 or num == 1:

return 1

else:

return num \* factorial(num - 1)

# Input from user

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

print(f"The factorial of {num} is {factorial(num)}.")

**2. Display the Multiplication Table**

# Function to display multiplication table

def multiplication\_table(num):

for i in range(1, 11):

print(f"{num} x {i} = {num \* i}")

# Input from user

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

multiplication\_table(num)

**3. Print the Fibonacci Sequence**

# Function to print the Fibonacci sequence

def fibonacci(n):

a, b = 0, 1

for \_ in range(n):

print(a, end=" ")

a, b = b, a + b

# Input from user

n = int(input("Enter the number of terms in the Fibonacci sequence: "))

fibonacci(n)

**4. Check Armstrong Number**

# Function to check Armstrong number

def check\_armstrong(num):

digits = len(str(num))

sum\_of\_digits = sum(int(digit) \*\* digits for digit in str(num))

return sum\_of\_digits == num

# Input from user

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

if check\_armstrong(num):

print(f"{num} is an Armstrong number.")

else:

print(f"{num} is not an Armstrong number.")

**5. Find Armstrong Numbers in an Interval**

# Function to check Armstrong number

def check\_armstrong(num):

digits = len(str(num))

sum\_of\_digits = sum(int(digit) \*\* digits for digit in str(num))

return sum\_of\_digits == num

# Input from user

start = int(input("Enter the starting number of the interval: "))

end = int(input("Enter the ending number of the interval: "))

print(f"Armstrong numbers between {start} and {end} are:")

for num in range(start, end + 1):

if check\_armstrong(num):

print(num, end=" ")

**6. Find the Sum of Natural Numbers**

# Function to find the sum of natural numbers

def sum\_of\_natural\_numbers(n):

return (n \* (n + 1)) // 2

# Input from user

n = int(input("Enter the value of n: "))

print(f"The sum of the first {n} natural numbers is {sum\_of\_natural\_numbers(n)}.")