**Name:- Milind Kailas Tajane**

**Roll No:- CS061**

**Date:-\_\_\_\_\_\_\_\_\_\_\_\_**

**Practical No:6**

# ----------------------------------------------------------------

**AIM:- Write a python program to find sum & multiplication of two matrices implemented using list.**

**---------------------------------------------------------------------------------------------------------------**

**CODE:-**

def add\_matrices(matrix1, matrix2):

*# Check if dimensions are the same*

*if* len(matrix1) *!=* len(matrix2) *or* len(matrix1[0]) *!=* len(matrix2[0]):

*raise* ValueError("Matrices must have the same dimensions for addition.")

    result *=* []

*for* i *in* range(len(matrix1)):

        row *=* []

*for* j *in* range(len(matrix1[0])):

            row.append(matrix1[i][j] *+* matrix2[i][j])

        result.append(row)

*return* result

def multiply\_matrices(matrix1, matrix2):

*# Check if matrices can be multiplied*

*if* len(matrix1[0]) *!=* len(matrix2):

*raise* ValueError("Number of columns in the first matrix must be equal to the number of rows in the second matrix.")

    result *=* []

*for* i *in* range(len(matrix1)):

        row *=* []

*for* j *in* range(len(matrix2[0])):

            sum\_product *=* 0

*for* k *in* range(len(matrix2)):

                sum\_product *+=* matrix1[i][k] *\** matrix2[k][j]

            row.append(sum\_product)

        result.append(row)

*return* result

*# Example usage:*

matrix1 *=* [

    [1, 2, 3],

    [4, 5, 6],

]

matrix2 *=* [

    [7, 8, 9],

    [1, 2, 3],

]

*# Matrix Addition*

sum\_result *=* add\_matrices(matrix1, matrix2)

print("Sum of matrices:")

*for* row *in* sum\_result:

    print(row)

*# Matrix Multiplication*

matrix3 *=* [

    [1, 2],

    [3, 4],

    [5, 6],

]

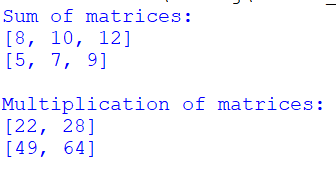
multiplication\_result *=* multiply\_matrices(matrix1, matrix3)

print("\nMultiplication of matrices:")

*for* row *in* multiplication\_result:

    print(row)

**Output:-**



**Name:- Milind Kailas Tajane**

**Roll No:- CS061**

**Date:-\_\_\_\_\_\_\_\_\_\_\_\_**

**Practical No:7**

# ----------------------------------------------------------------

**AIM:- Write a program to implement function decorator to display cube of a number.**

**----------------------------------------------------------------------------------------------------------------**

**CODE:-**

# Define the decorator

def cube\_decorator(func):

def wrapper(num):

result = func(num)

print(f"The cube of {num} is {result}")

return result

return wrapper

# Use the decorator on a function

@cube\_decorator

def cube(num):

return num \*\* 3

# Test the function

if \_\_name\_\_ == "\_\_main\_\_":

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

cube(number)

**Output:-**



**Name:- Milind Kailas Tajane**

**Roll No:- CS061**

**Date:-\_\_\_\_\_\_\_\_\_\_\_\_**

**Practical No: 8**

# ----------------------------------------------------------------

**AIM:- Write a program to implement generator function to display square of numbers from 1 to 100 .**

**----------------------------------------------------------------------------------------------------------------**

**CODE:-**

# Define the generator function

def square\_generator():

for num in range(1, 11):

yield num \*\* 2

# Use the generator function

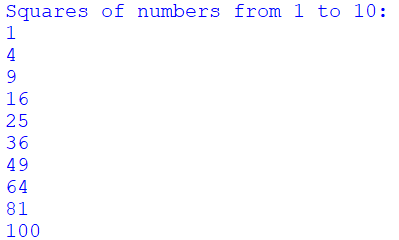
if \_\_name\_\_ == "\_\_main\_\_":

print("Squares of numbers from 1 to 10:")

for square in square\_generator():

print(square)

**Output:-**



**Name:- Milind Kailas Tajane**

**Roll No:- CS061**

**Date:-\_\_\_\_\_\_\_\_\_\_\_\_**

**Practical No: 9**

# ----------------------------------------------------------------

**AIM:- Write a program to count even and odd number in the list.**

**----------------------------------------------------------------------------------------------------------------**

**CODE:-**

# Function to count even and odd numbers in a list

def count\_even\_odd(numbers):

# Initialize counters for even and odd numbers

even\_count = 0

odd\_count = 0

# Iterate through each number in the provided list

for number in numbers:

# Check if the number is even

if number % 2 == 0:

even\_count += 1 # Increment even count

else:

odd\_count += 1 # Increment odd count

# Return the counts of even and odd numbers

return even\_count, odd\_count

# List of numbers from 1 to 20

numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20]

# Call the function and store the results in even\_count and odd\_count

even\_count, odd\_count = count\_even\_odd(numbers)

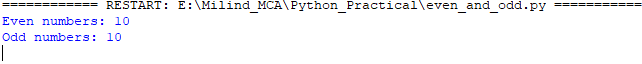
# Print the count of even numbers

print(f"Even numbers: {even\_count}")

# Print the count of odd numbers

print(f"Odd numbers: {odd\_count}")

**Output:-**



**Name:- Milind Kailas Tajane**

**Roll No:- CS061**

**Date:-\_\_\_\_\_\_\_\_\_\_\_\_**

**Practical No: 5**

# ----------------------------------------------------------------

**AIM:- Write a program to find sum of all numbers, mean, max, average of numbers in a list.**

**----------------------------------------------------------------------------------------------------------------**

**CODE:-**

from collections import Counter

def calculate\_statistics(numbers):

if not numbers:

return None, None, None, None, None

# Calculate sum

total\_sum = sum(numbers)

# Calculate minimum

minimum = min(numbers)

# Calculate maximum

maximum = max(numbers)

# Calculate mean

mean = total\_sum / len(numbers)

# Calculate mode

frequency = Counter(numbers)

mode\_data = frequency.most\_common()

mode = [num for num, freq in mode\_data if freq == mode\_data[0][1]]

return total\_sum, minimum, maximum, mean, mode

# Example usage

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

total\_sum, minimum, maximum, mean, mode = calculate\_statistics(numbers)

print(f"Sum: {total\_sum}")

print(f"Min: {minimum}")

print(f"Max: {maximum}")

print(f"Mean: {mean}")

print(f"Mode: {mode}")

**Output:-**

