**PYTHON DAY-4**

**1.PERFECT NO**

def is\_perfect\_number(n):

return n == sum(i for i in range(1, n) if n % i == 0)

num = 28

if is\_perfect\_number(num):

print(f"{num} is a perfect number")

else:

print(f"{num} is not a perfect number")

**2.TRANSVERSE OF MATRIX**

def transpose\_matrix(matrix):

return [list(row) for row in zip(\*matrix)]

matrix = [

[1, 2, 3],

[4, 5, 6],

[7, 8, 9]

]

transposed = transpose\_matrix(matrix)

for row in transposed:

print(row)

**3.DIAGONAL SUM OF MATRICES**

def matrix\_sums(matrix):

n = len(matrix)

row\_sums = [sum(row) for row in matrix]

col\_sums = [sum(matrix[i][j] for i in range(n)) for j in range(n)]

main\_diag\_sum = sum(matrix[i][i] for i in range(n))

anti\_diag\_sum = sum(matrix[i][n - 1 - i] for i in range(n))

return row\_sums, col\_sums, main\_diag\_sum, anti\_diag\_sum

matrix = [

[1, 2, 3],

[4, 5, 6],

[7, 8, 9]

]

row\_sums, col\_sums, main\_diag\_sum, anti\_diag\_sum = matrix\_sums(matrix)

print("Row sums:", row\_sums)

print("Column sums:", col\_sums)

print("Main diagonal sum:", main\_diag\_sum)

print("Anti-diagonal sum:", anti\_diag\_sum)

**4.SUM OF BOUNDARY ELEMENT**

def sum\_boundary\_elements(matrix):

if not matrix or not matrix[0]:

return 0

return sum(matrix[0] + matrix[-1]) + sum(row[0] + row[-1] for row in matrix[1:-1])

matrix = [

[1, 2, 3],

[4, 5, 6],

[7, 8, 9]

]

boundary\_sum = sum\_boundary\_elements(matrix)

print("Sum of boundary elements:", boundary\_sum)

**5.SPIRAL ORDER**

def spiral\_order(matrix):

result = []

while matrix:

result += matrix.pop(0)

matrix = list(zip(\*matrix))[::-1]

return result

matrix = [

[1, 2, 3],

[4, 5, 6],

[7, 8, 9]

]

spiral = spiral\_order(matrix)

print("Spiral order:", spiral)

**6.SUM OF N NUMBERS**

def sum\_n\_numbers(n):

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

n = 10

total = sum\_n\_numbers(n)

print("Sum of first", n, "numbers:", total)

**7.SUM OF 1 FAC+2 FAC**

def sum\_of\_factorials(n):

from math import factorial

return sum(factorial(i) for i in range(1, n + 1))

n = 5

total = sum\_of\_factorials(n)

print("Sum of factorials from 1 to", n, ":", total)

**8.FIND SUM OF 1^2+2^2+….N^2**

def sum\_of\_squares(n):

total = 0

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

total += i \*\* 2

return total

n = 5

print(f"The sum of squares of the first {n} natural numbers is: {sum\_of\_squares(n)}")

**9.MEAN,MEDIAN AND MODE**

from statistics import mean, median, mode

def stats\_summary(data):

return {

'mean': mean(data),

'median': median(data),

'mode': mode(data)

}

data = [1, 2, 2, 3, 4]

stats = stats\_summary(data)

print("Mean:", stats['mean'])

print("Median:", stats['median'])

print("Mode:", stats['mode'])

**10.N LARGEST NUMBER**

def find\_nth\_largest(numbers, n):

sorted\_numbers = sorted(numbers, reverse=True)

return sorted\_numbers[n-1]

numbers = [3, 1, 4, 1, 5, 9, 2, 6, 5, 3, 5]

n = 3

print(f"The {n}rd largest number is: {find\_nth\_largest(numbers, n)}")