**Python Programming :**

**CSA0818 :**

**NAME: Nanda Kishor K**

**Rg.No:192311148**

**# 1.Perfect number :-**

**Code:**

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

sum\_of\_divisors = 0

for i in range(1, number):

if number % i == 0:

sum\_of\_divisors += i

if sum\_of\_divisors == number:

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

else:

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

**Input :** Enter a number: 28

**Output :** 28 is a perfect number.

**# 2.Transpose of given matrix :-**

**Code:**

matrix = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]

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

for row in transpose:

print(row)

**Output :** [1, 4, 7]

[2, 5, 8]

[3, 6, 9]

**# 3.To find row ,column and diagonal sum of given matrix:-**

**Code:**

matrix]

colsums = [sum(matrix[i][j] for i in matrix = [

[1, 2, 3],

[4, 5, 6],

[7, 8, 9]

]

n = len(matrix)

rowsums = [sum(row) for row in range(n)) for j in range(n)]

primarydiagsum = sum(matrix[i][i] for i in range(n))

secondarydiagsum = sum(matrix[i][n-1-i] for i in range(n))

print(f"Row sums: {rowsums}")

print(f"Column sums: {colsums}")

print(f"Primary diagonal sum: {primarydiagsum}")

print(f"Secondary diagonal sum: {secondarydiagsum}")

**Output :** Row sums: [6, 15, 24]

Column sums: [12, 15, 18]

Primary diagonal sum: 15

Secondary diagonal sum: 15

**# 4.Sum of boundary elements:-**

**Code:**

matrix = [

[1, 2, 3],

[4, 5, 6],

[7, 8, 9]

]

n, m = len(matrix), len(matrix[0])

boundarysum = sum(matrix[0] + matrix[-1]) + sum(matrix[i][0] + matrix[i][-1] for i in range(1, n-1))

print(f"Sum of boundary elements: {boundarysum}")

**Output :** Sum of boundary elements: 40

**# 5.Matrix in spiral order:-**

**Code:**

matrix = [

[1, 2, 3, 4],

[5, 6, 7, 8],

[9, 10, 11, 12],

[13, 14, 15, 16]

]

top = 0

bottom = len(matrix) - 1

left = 0

right = len(matrix[0]) - 1

while top <= bottom and left <= right:

for i in range(left, right + 1):

print(matrix[top][i], end=' ')

top += 1

for i in range(top, bottom + 1):

print(matrix[i][right], end=' ')

right -= 1

if top <= bottom:

for i in range(right, left - 1, -1):

print(matrix[bottom][i], end=' ')

bottom -= 1

if left <= right:

for i in range(bottom, top - 1, -1):

print(matrix[i][left], end=' ')

left += 1

print()

**Output :** 1 2 3 4 8 12 16 15 14 13 9 5 6 7 11 10

**# 6.Sum of n numbers:-**

**Code:**

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

sum = n \* (n + 1) // 2

print(f"Sum of the first {n} numbers is: {sum}")

**Input :** Enter a number: 5

**Output :** Sum of the first 5 numbers is: 15

**# 7.Sum of n factorial:-**

**Code:**

import math

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

sum = sum(math.factorial(i) for i in range(1, n + 1))

print(f"Sum of factorials from 1! to {n}! is: {sum}")

**Input :** Enter a number: 5

**Output :** Sum of factorials from 1! to 5! is: 153

**# 8.Sum of squares:-**

**Code:**

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

sum = n \* (n + 1) \* (2 \* n + 1) // 6

print(f"Sum of squares from 1^2 to {n}^2 is: {sum}")

**Input :** Enter a number: 5

**Output :** Sum of squares from 1^2 to 5^2 is: 55

**# 9. Mean , Median and Mode of element :-**

**Code:**

from statistics import mean, median, mode, StatisticsError

data = [1, 2, 2, 3, 4, 4, 4, 5, 6]

mean\_val = mean(data)

median\_val = median(data)

try:

mode\_val = mode(data)

except StatisticsError:

mode\_val = "No unique mode"

print(f"Mean: {mean\_val}\nMedian: {median\_val}\nMode: {mode\_val}")

**Output :** Mean: 3.4444444444444446

Median: 4

Mode: 4

**# 10.nth largest number in the given list :-**

**Code:**

data = [10, 20, 4, 45, 99]

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

if n > len(data) or n <= 0:

print("Invalid value for n")

else:

data.sort(reverse=True)

print(f"The {n}th largest number is: {data[n - 1]}")

**Input :** Enter the value of n: 3

**Output :** The 3th largest number is: 20