1. Find the Mth maximum number and Nth minimum number in an array and then find the sum of it and difference of it.

Sample Input:

Array of elements = {14, 16, 87, 36, 25, 89, 34}

M = 1

N = 3

Sample Output:

1st Maximum Number = 89

3rd Minimum Number = 25

Sum = 114

Difference = 64

**Test cases:**

A. {16, 16, 16 16, 16}, M = 0, N = 1

B. {0, 0, 0, 0}, M = 1, N = 2

C. {-12, -78, -35, -42, -85}, M = 3 , N = 3

D. {15, 19, 34, 56, 12}, M = 6 , N = 3

E. {85, 45, 65, 75, 95}, M = 5 , N = 7

**PROGRAM:**

def find\_mth\_max\_nth\_min(arr, m, n):

arr\_sorted = sorted(arr)

mth\_max = arr\_sorted[-m] if m <= len(arr) else None

nth\_min = arr\_sorted[n-1] if n <= len(arr) else None

if mth\_max is None or nth\_min is None:

return "Invalid input for M or N"

sum\_val = mth\_max + nth\_min

diff\_val = mth\_max - nth\_min

return (f"{m}st Maximum Number = {mth\_max}\n"

f"{n}rd Minimum Number = {nth\_min}\n"

f"Sum = {sum\_val}\n"

f"Difference = {diff\_val}")

test\_cases = [

([14, 16, 87, 36, 25, 89, 34], 1, 3),

([16, 16, 16, 16, 16], 0, 1),

([0, 0, 0, 0], 1, 2),

([-12, -78, -35, -42, -85], 3, 3),

([15, 19, 34, 56, 12], 6, 3),

([85, 45, 65, 75, 95], 5, 7)

]

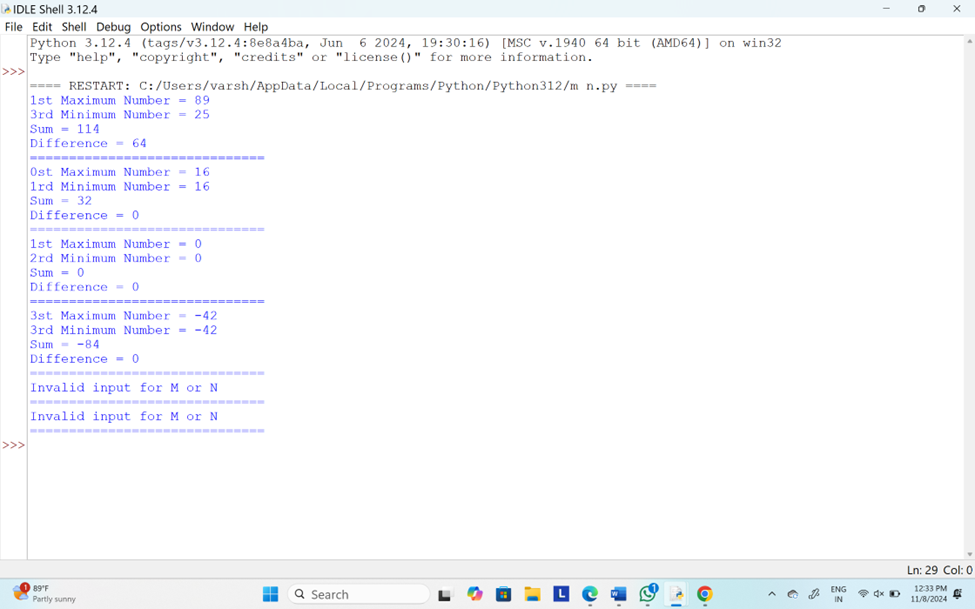
for case in test\_cases:

arr, m, n = case

print(find\_mth\_max\_nth\_min(arr, m, n))

print("=" \* 30)

**OUTPUT:**

****

2. Write a program to count all the prime and composite numbers entered by the user.

Sample Input:

Enter the numbers

4

54

29

71

7

59

98

23

Sample Output:

Composite number:3

Prime number:5

**Test cases:**

A. 33, 41, 52, 61,73,90

B. TEN, FIFTY, SIXTY-ONE, SEVENTY-SEVEN, NINE

C. 45, 87, 09, 5.0 ,2.3, 0.4

D. -54, -76, -97, -23, -33, -98

E. 45, 73, 00, 50, 67, 44

**PROGRAM:**

def is\_prime(num):

"""Function to check if a number is prime."""

if num <= 1:

return False

for i in range(2, int(num \*\* 0.5) + 1):

if num % i == 0:

return False

return True

def is\_composite(num):

"""Function to check if a number is composite."""

return num > 1 and not is\_prime(num)

def count\_prime\_composite(numbers):

"""Function to count the prime and composite numbers."""

prime\_count = 0

composite\_count = 0

for num in numbers:

if isinstance(num, int) and num >= 0: # Checking if the number is a valid non-negative integer

if is\_prime(num):

prime\_count += 1

elif is\_composite(num):

composite\_count += 1

return prime\_count, composite\_count

def main():

# Taking user input for numbers

user\_input = input("Enter the numbers separated by space: ")

# Split the input and try to convert to integers

try:

# Filtering out non-numeric values and invalid inputs

numbers = [int(x) for x in user\_input.split() if x.isdigit()]

prime\_count, composite\_count = count\_prime\_composite(numbers)

print(f"Composite number: {composite\_count} Prime number: {prime\_count}")

except ValueError:

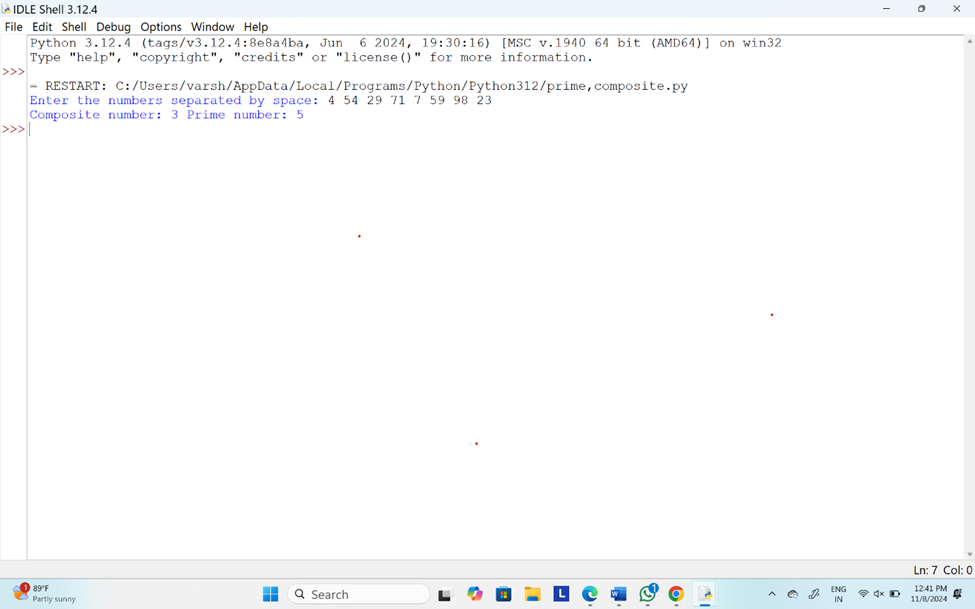
print("Invalid input. Please enter only integers.")

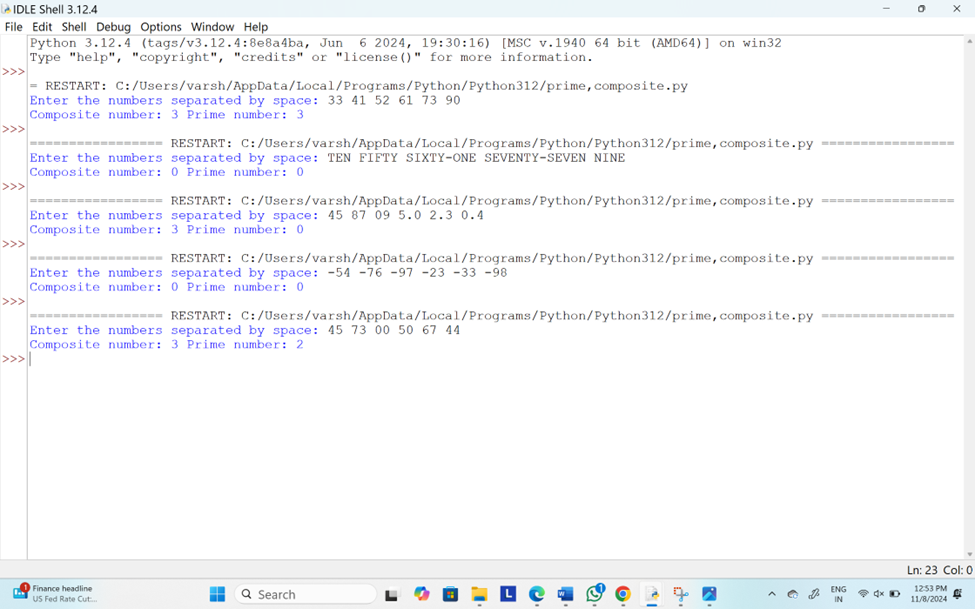
# Run the main function

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

main()

**OUTPUT:**

****

****

3.Write the program to display most and least significant digit of a number

**PROGRAM:**

def get\_most\_and\_least\_significant\_digit(number):

# Ensure the number is a positive integer

if number < 0:

number = abs(number)

# Convert the number to a string for easy access to digits

number\_str = str(number)

# Least Significant Digit (LSD) is the last digit of the string

lsd = number\_str[-1]

# Most Significant Digit (MSD) is the first non-zero digit

msd = number\_str[0]

return msd, lsd

def main():

# Taking input from the user

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

msd, lsd = get\_most\_and\_least\_significant\_digit(number)

print(f"Most Significant Digit: {msd}")

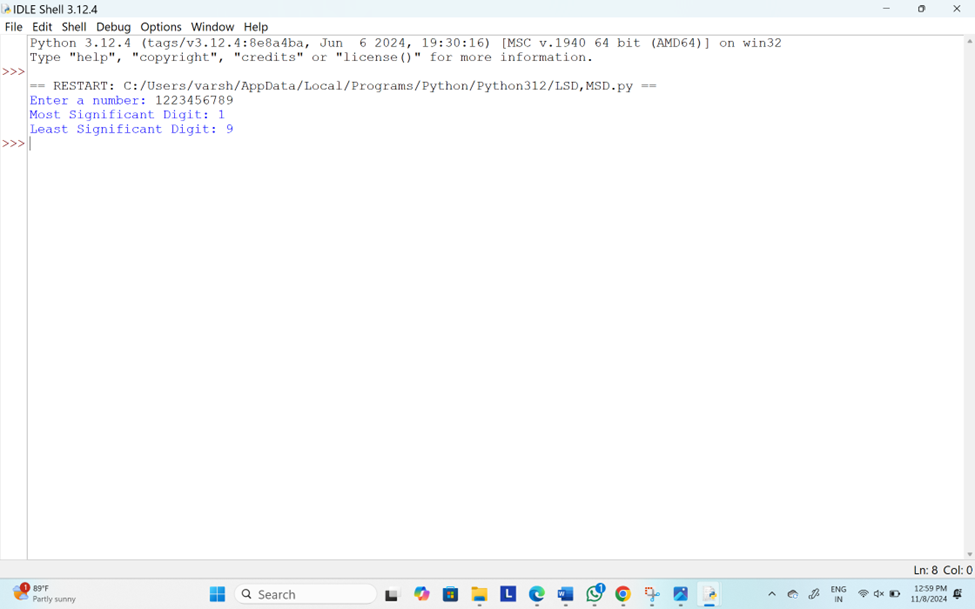
print(f"Least Significant Digit: {lsd}")

# Run the program

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

main()

**OUTPUT:**

****

4.Find the program to print the prime numbers between the range from M to N

**PROGRAM:**

def is\_prime(num):

"""Function to check if a number is prime."""

if num <= 1:

return False

for i in range(2, int(num \*\* 0.5) + 1):

if num % i == 0:

return False

return True

def print\_primes\_in\_range(m, n):

"""Function to print all prime numbers in the range [m, n]."""

for num in range(m, n + 1):

if is\_prime(num):

print(num, end=" ")

def main():

# Taking user input for the range

m = int(input("Enter the starting number (M): "))

n = int(input("Enter the ending number (N): "))

print(f"Prime numbers between {m} and {n} are:")

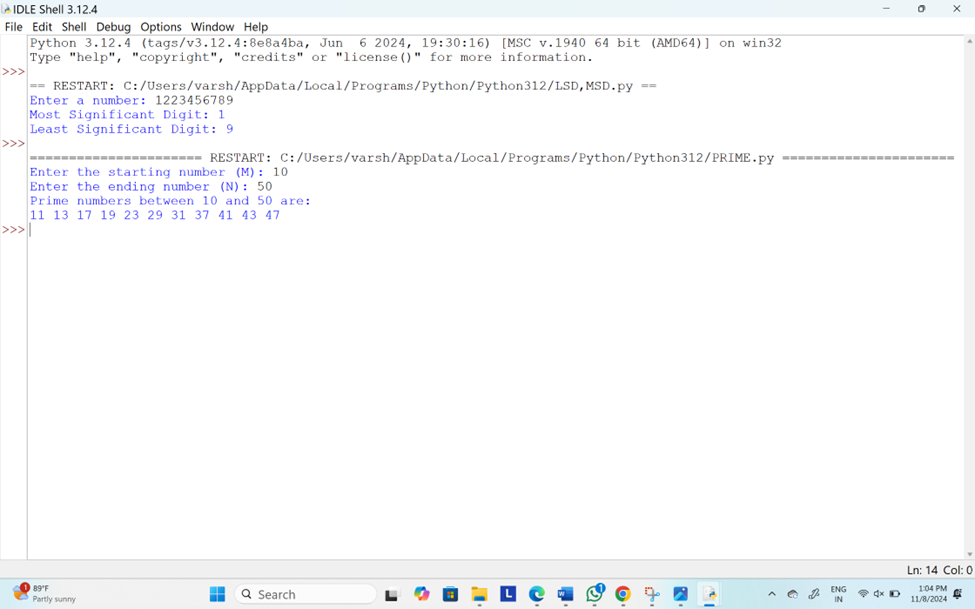
print\_primes\_in\_range(m, n)

# Run the program

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

main()

**OUTPUT:**

****

5.Write the program to check whether a given year is a leap year or not

**PROGRAM:**

def is\_leap\_year(year):

"""Function to check if a given year is a leap year."""

# A year is a leap year if:

# 1. It is divisible by 4, but not divisible by 100, OR

# 2. It is divisible by 400.

if (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0):

return True

else:

return False

def main():

# Taking input from the user

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

# Check and display if the year is a leap year

if is\_leap\_year(year):

print(f"{year} is a leap year.")

else:

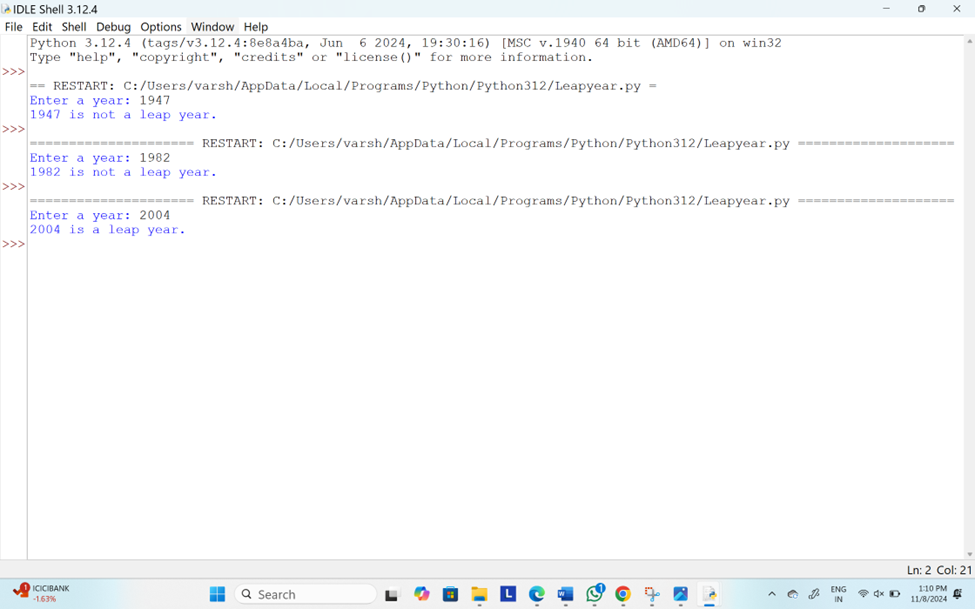
print(f"{year} is not a leap year.")

# Run the program

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

main()

**OUTPUT:**

****

6.Find the program to the integer n,return the least number of perfect square numbers that sum to n.A perfect square is an integer that is the square of an integer,in other words,it is the product of some integer itself.For example 1,4,9 and 16 are perfect squares while 3 and 11 are not

**PROGRAM:**

import math

def numSquares(n):

# Create a DP array where dp[i] represents the least number of perfect squares that sum to i

dp = [float('inf')] \* (n + 1)

dp[0] = 0 # Base case: 0 perfect squares are needed to sum to 0

# List of perfect squares less than or equal to n

squares = [i \* i for i in range(1, int(math.sqrt(n)) + 1)]

# Fill the DP array

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

for square in squares:

if i >= square:

dp[i] = min(dp[i], dp[i - square] + 1)

return dp[n]

# Main function to take input and display the result

def main():

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

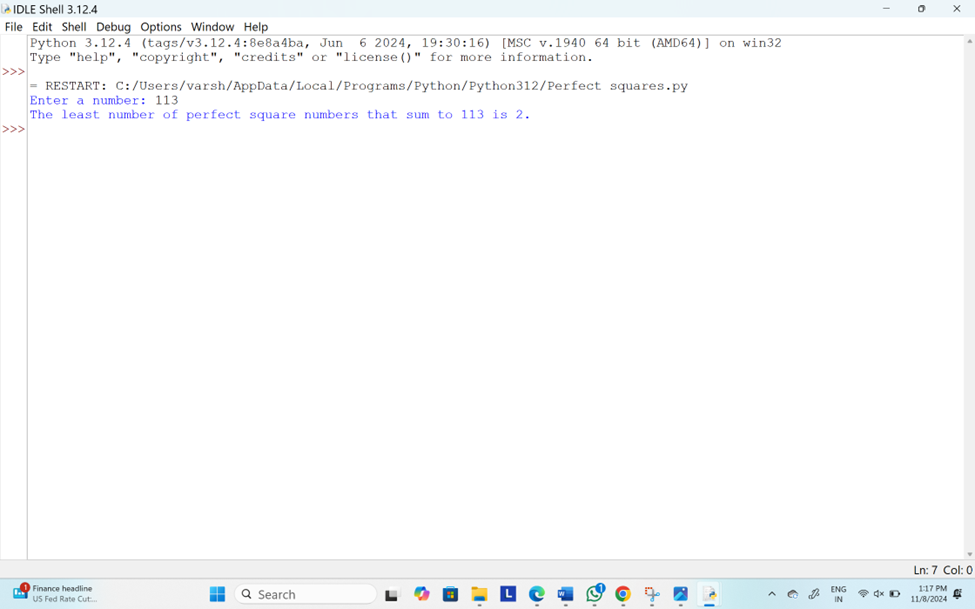
print(f"The least number of perfect square numbers that sum to {n} is {numSquares(n)}.")

# Run the program

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

main()

**OUTPUT:**

****

7.Write the program to print multiplication of 3 matrices

**PROGRAM:**

def multiply\_matrices(A, B):

"""Multiplies two matrices A and B."""

# Number of rows in A and number of columns in B

rows\_A = len(A)

cols\_A = len(A[0])

rows\_B = len(B)

cols\_B = len(B[0])

# Check if multiplication is possible (cols of A must be equal to rows of B)

if cols\_A != rows\_B:

raise ValueError("Matrices cannot be multiplied due to incompatible dimensions.")

# Initialize the result matrix with zeros

result = [[0 for \_ in range(cols\_B)] for \_ in range(rows\_A)]

# Matrix multiplication

for i in range(rows\_A):

for j in range(cols\_B):

for k in range(cols\_A):

result[i][j] += A[i][k] \* B[k][j]

return result

def multiply\_three\_matrices(A, B, C):

"""Multiplies three matrices A, B, and C."""

# First, multiply A and B

AB = multiply\_matrices(A, B)

# Then multiply the result with C

ABC = multiply\_matrices(AB, C)

return ABC

def print\_matrix(matrix):

"""Helper function to print a matrix."""

for row in matrix:

print(row)

def main():

# Input matrices A, B, and C

print("Enter the dimensions of matrix A (rows x columns):")

rows\_A, cols\_A = map(int, input().split())

print("Enter the elements of matrix A:")

A = [list(map(int, input().split())) for \_ in range(rows\_A)]

print("Enter the dimensions of matrix B (rows x columns):")

rows\_B, cols\_B = map(int, input().split())

print("Enter the elements of matrix B:")

B = [list(map(int, input().split())) for \_ in range(rows\_B)]

print("Enter the dimensions of matrix C (rows x columns):")

rows\_C, cols\_C = map(int, input().split())

print("Enter the elements of matrix C:")

C = [list(map(int, input().split())) for \_ in range(rows\_C)]

# Check if matrix multiplication is possible

if cols\_A != rows\_B or cols\_B != rows\_C:

print("Matrix multiplication is not possible due to incompatible dimensions.")

return

# Multiply the three matrices

result = multiply\_three\_matrices(A, B, C)

# Print the result

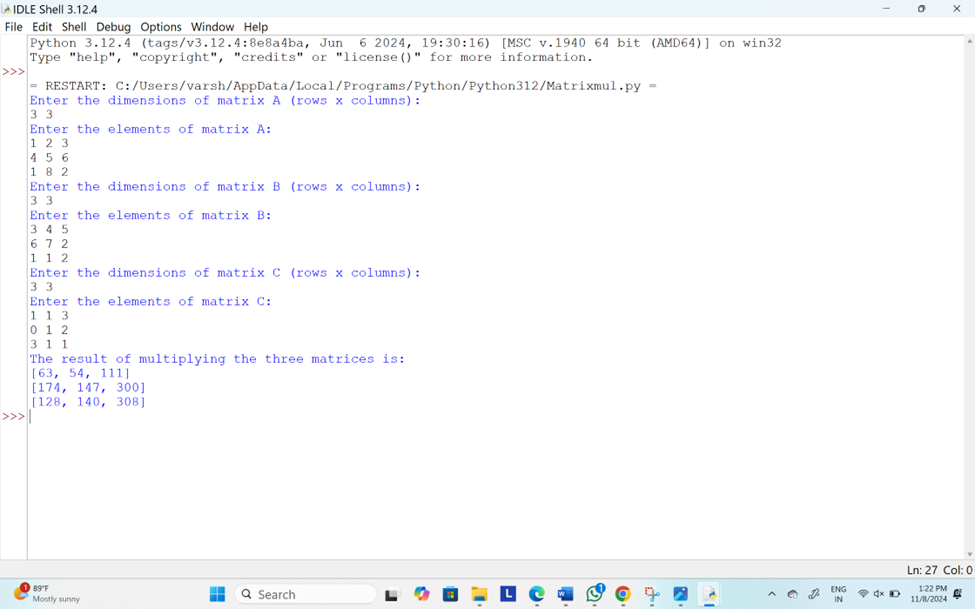
print("The result of multiplying the three matrices is:")

print\_matrix(result)

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

main()

**OUTPUT:**

****

8.Find the program array arr which contains integer elements sort these elements in ascending order if any negative number is found it has to be replaced with the average of an array

**PROGRAM:**

def replace\_negatives\_with\_average(arr):

"""Function to replace negative numbers with the average of the array."""

# Calculate the average of the array

if len(arr) == 0:

return arr # Return empty array if no elements exist

avg = sum(arr) / len(arr)

# Replace negative numbers with the average

for i in range(len(arr)):

if arr[i] < 0:

arr[i] = avg

return arr

def main():

# Input array from the user

arr = list(map(int, input("Enter the elements of the array (space separated): ").split()))

# Replace negative numbers with the average

arr = replace\_negatives\_with\_average(arr)

# Sort the array in ascending order

arr.sort()

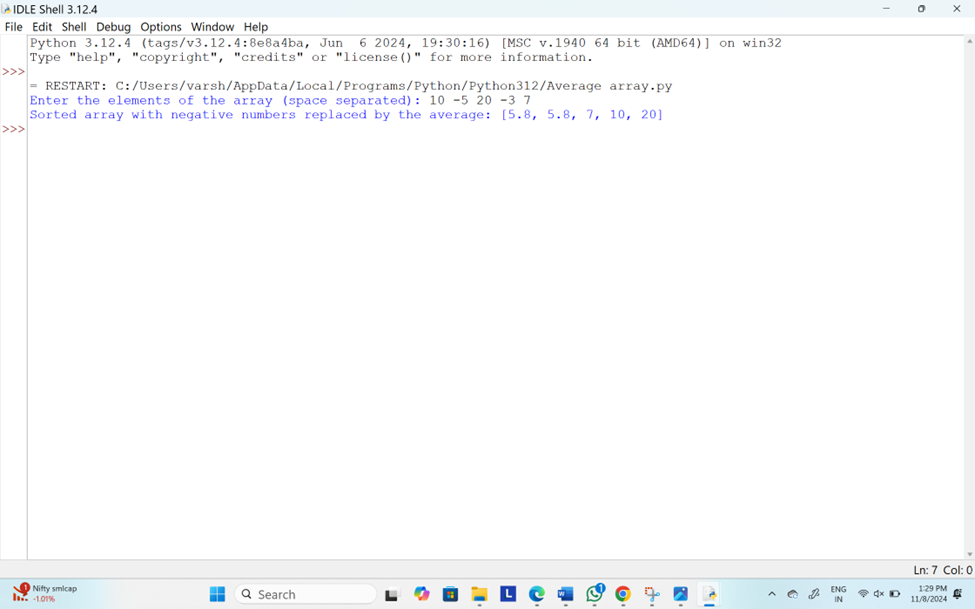
# Output the final sorted array

print("Sorted array with negative numbers replaced by the average:", arr)

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

main()

**OUTPUT:**

****