21.Write a program to print the first n perfect numbers. (Hint Perfect number means **a** positive integer that is equal to the sum of its proper divisors)

Sample Input:

N = 3

Sample Output:

First 3 perfect numbers are: 6 , 28 , 496

**Test Cases:**

1. N = 0
2. N = 5
3. N = -2
4. N = -5
5. N = 0.2

**PROGRAM:**

def sum\_of\_divisors(num):

"""Returns the sum of proper divisors of the given number."""

divisors\_sum = 0

for i in range(1, num // 2 + 1):

if num % i == 0:

divisors\_sum += i

return divisors\_sum

def is\_perfect\_number(num):

"""Check if a number is a perfect number."""

return sum\_of\_divisors(num) == num

def find\_perfect\_numbers(n):

"""Find and return the first n perfect numbers."""

perfect\_numbers = []

num = 2 # Start checking from 2 (since 1 is not a perfect number)

while len(perfect\_numbers) < n:

if is\_perfect\_number(num):

perfect\_numbers.append(num)

num += 1

return perfect\_numbers

def main():

try:

n = float(input("Enter the number of perfect numbers to print (N): "))

if n <= 0 or n != int(n): # Check if n is a non-positive or non-integer number

print("Please enter a valid positive integer.")

return

n = int(n) # Convert to integer

if n == 0:

print("No perfect numbers to print.")

return

perfect\_numbers = find\_perfect\_numbers(n)

print(f"First {n} perfect numbers are: {', '.join(map(str, perfect\_numbers))}")

except ValueError:

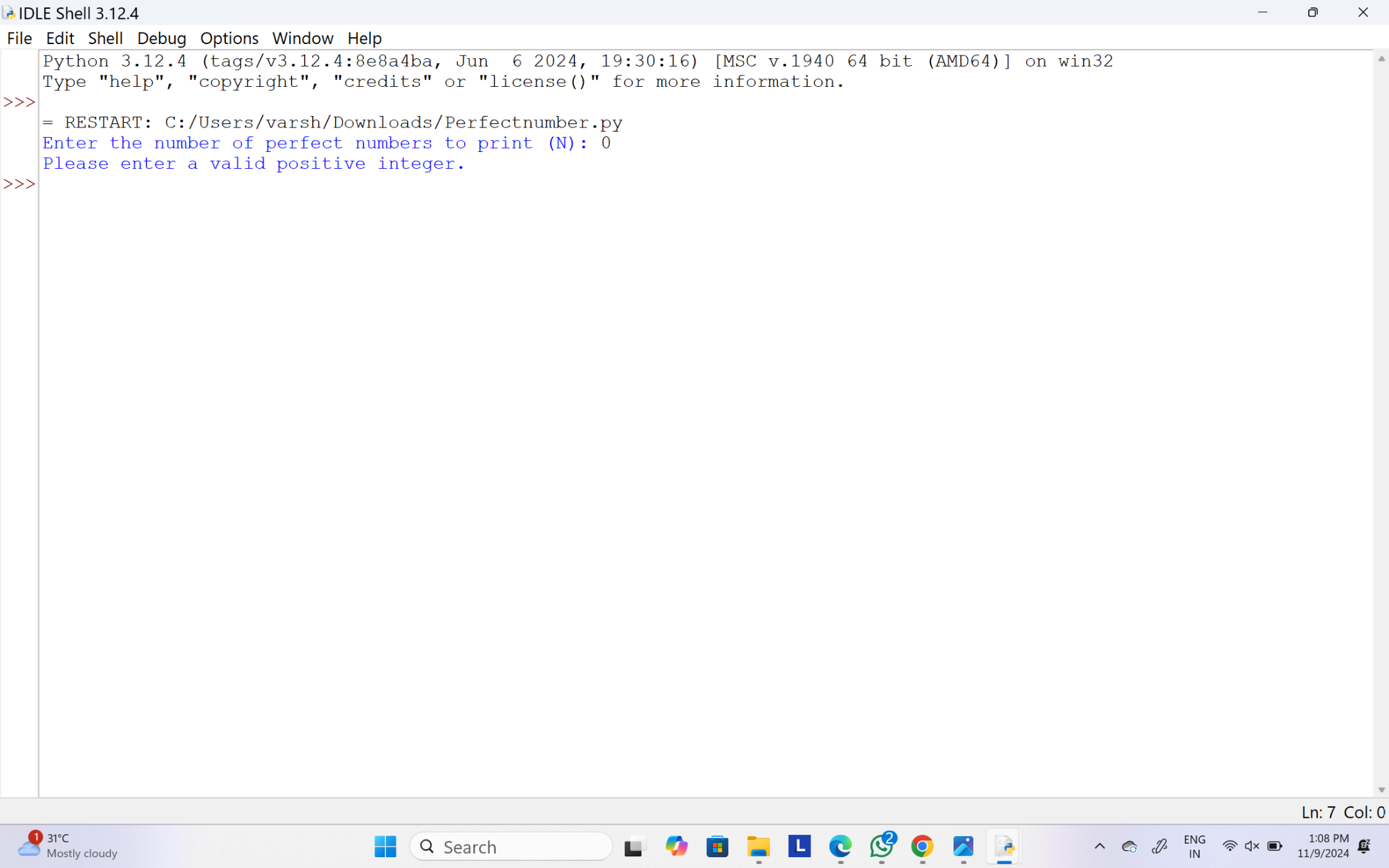
print("Invalid input. Please enter a valid positive integer.")

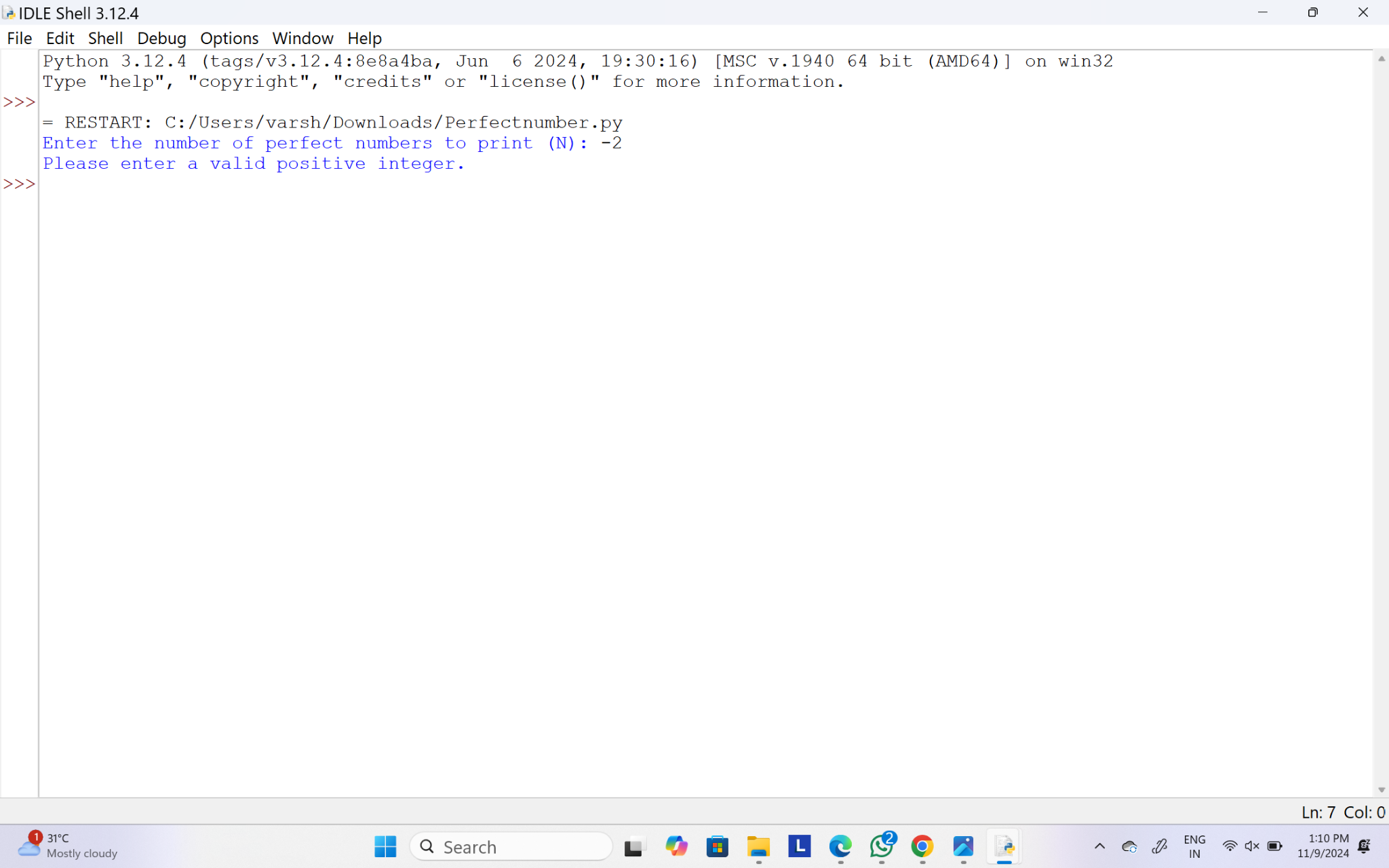
if \_\_name\_\_ == "\_\_main\_\_":

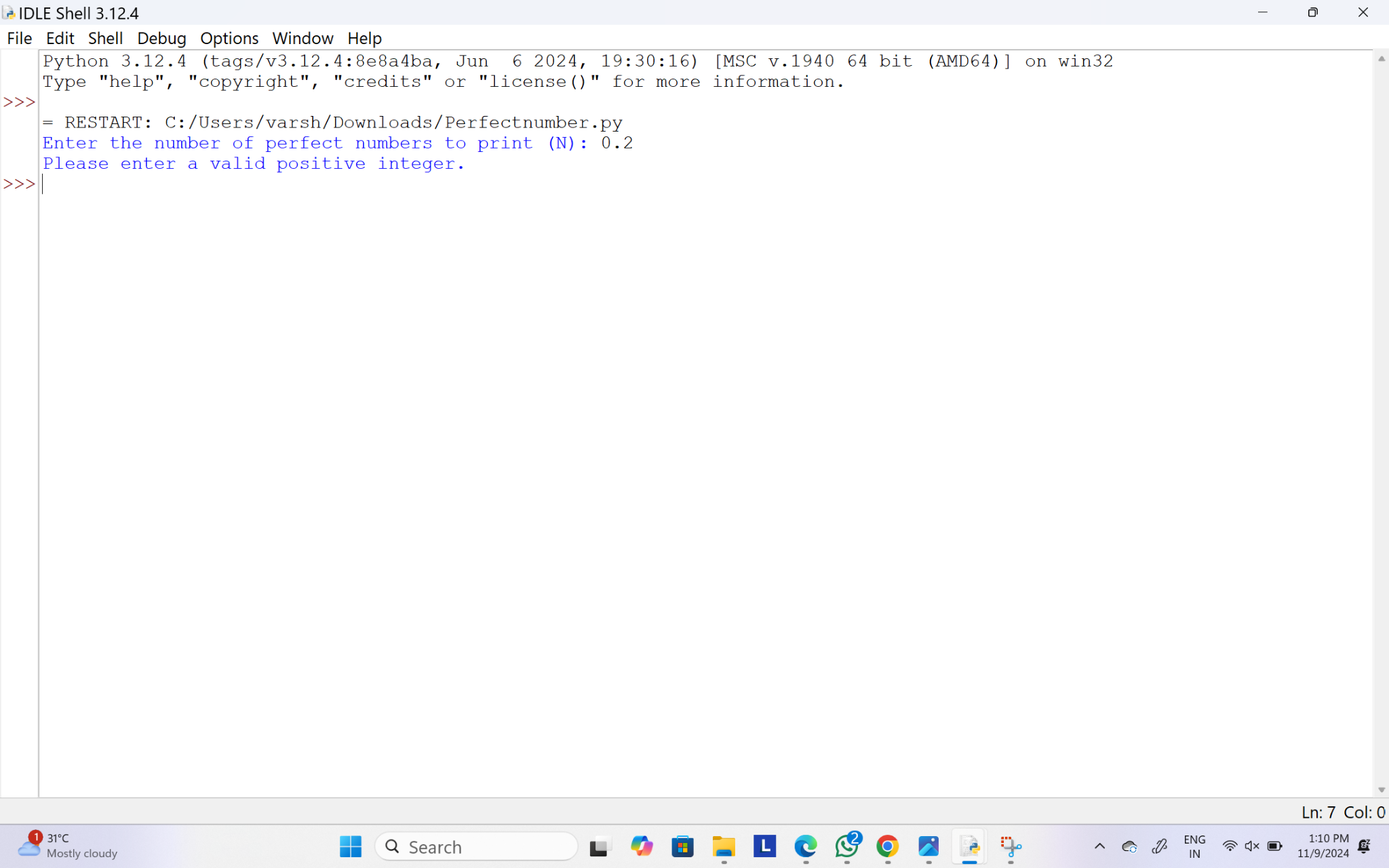
main()

**OUTPUT:**

****

****

****

****

22.Write a program to enter the marks of a student in four subjects. Then calculate the total and aggregate, display the grade obtained by the student. If the student scores an aggregate greater than 75%, then the grade is Distinction. If aggregate is 60>= and <75, then the grade is First Division. If aggregate is 50 >= and <60, then the grade is Second Division. If aggregate is 40>= and <50, then the grade is Third Division. Else the grade is Fail.

Sample Input & Output:

Enter the marks in python: 90

Enter the marks in c programming: 91

Enter the marks in Mathematics: 92

Enter the marks in Physics: 93

Total= 366

Aggregate = 91.5

DISTINCTION

**Test cases:**

1. 18, 76,93,65
2. 73,78,79,75
3. 98,106,120,95
4. 96,73, -85,95
5. 78,59.8,76,79

**PROGRAM:**

def calculate\_grade(marks):

total = sum(marks)

aggregate = total / 4

print(f"Total = {total}")

print(f"Aggregate = {aggregate:.2f}")

if aggregate > 75:

print("DISTINCTION")

elif 60 <= aggregate < 75:

print("FIRST DIVISION")

elif 50 <= aggregate < 60:

print("SECOND DIVISION")

elif 40 <= aggregate < 50:

print("THIRD DIVISION")

else:

print("FAIL")

def main():

try:

python\_marks = float(input("Enter the marks in python: "))

c\_marks = float(input("Enter the marks in c programming: "))

math\_marks = float(input("Enter the marks in Mathematics: "))

physics\_marks = float(input("Enter the marks in Physics: "))

marks = [python\_marks, c\_marks, math\_marks, physics\_marks]

calculate\_grade(marks)

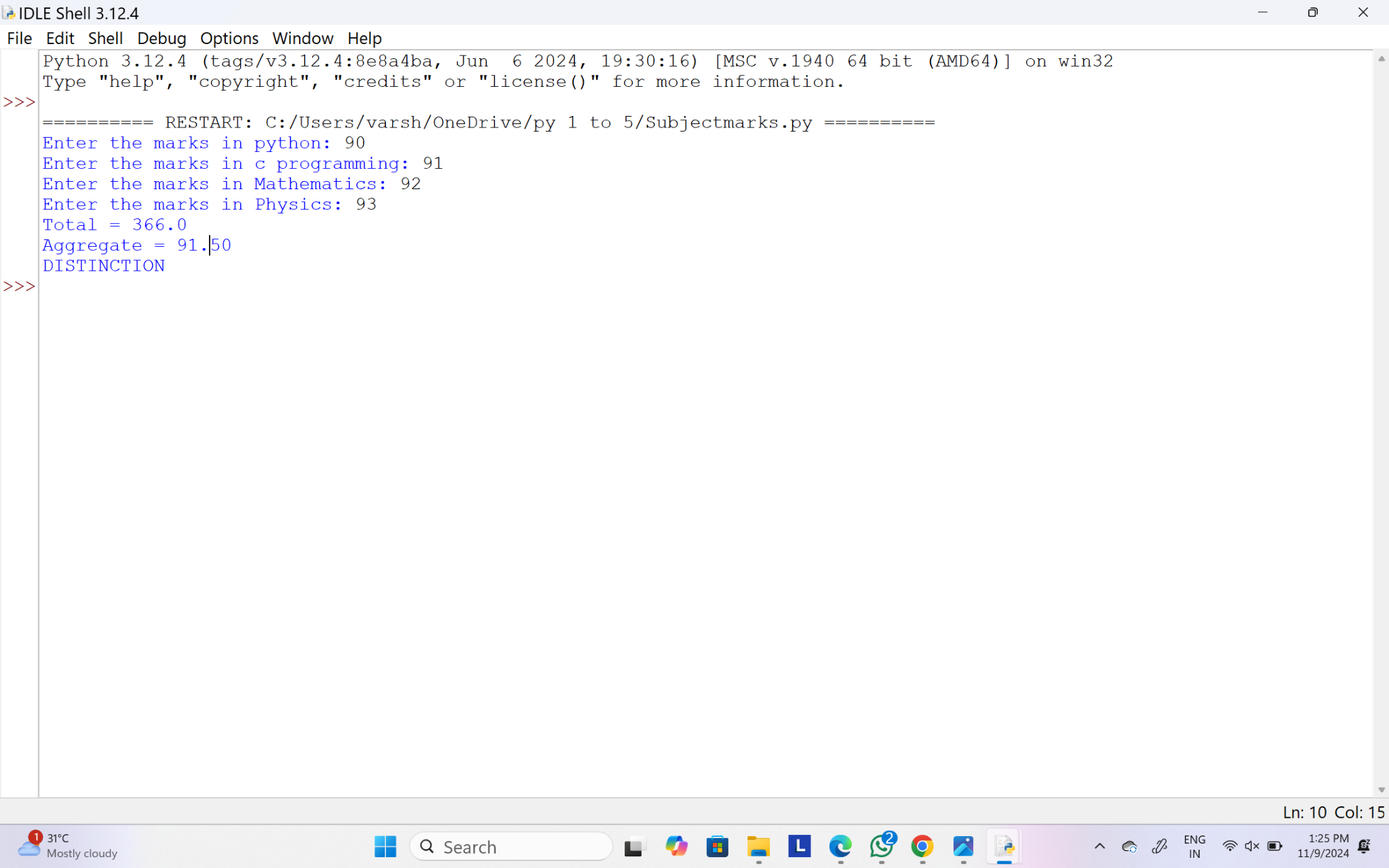
except ValueError:

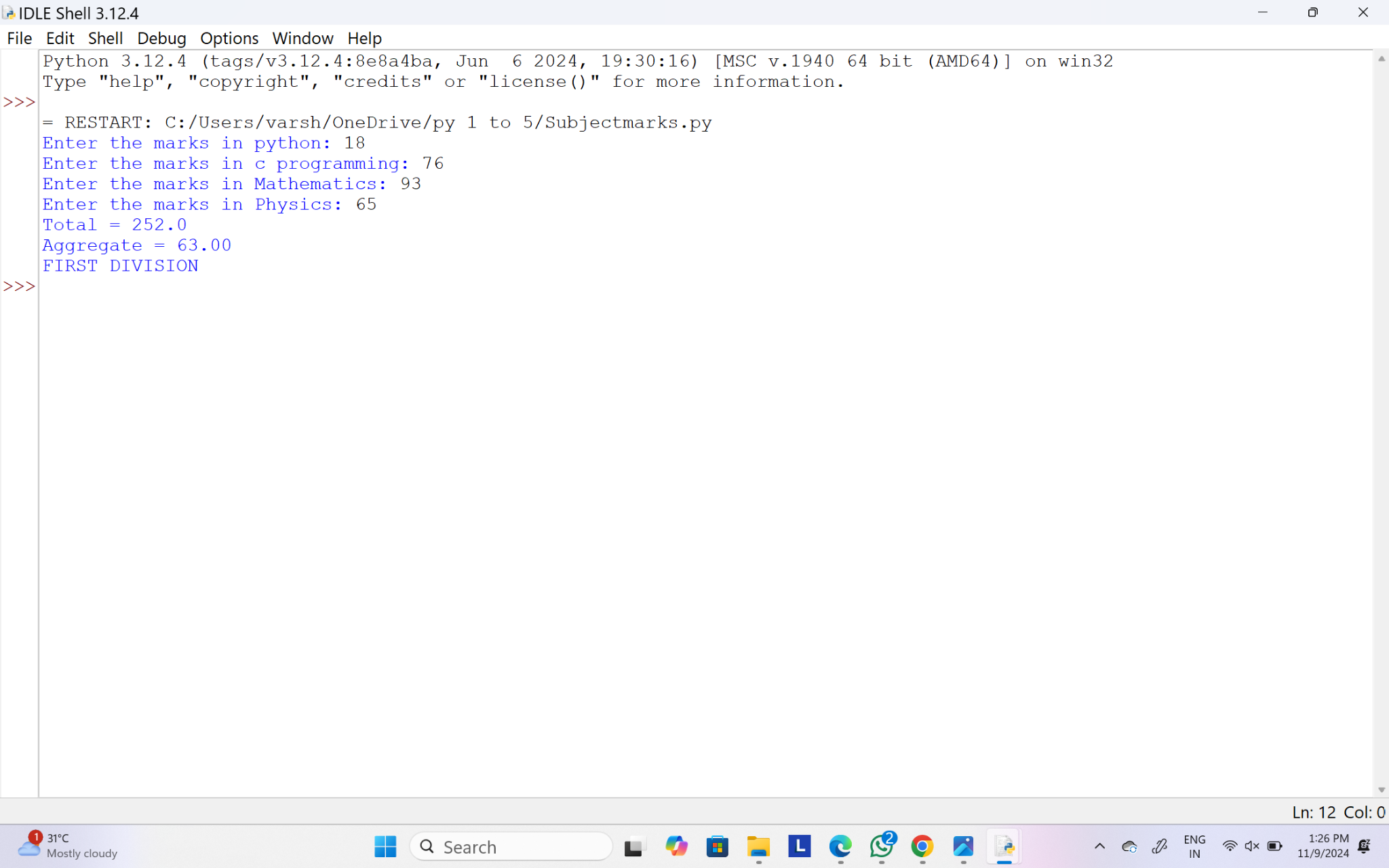
print("Please enter valid numeric marks.")

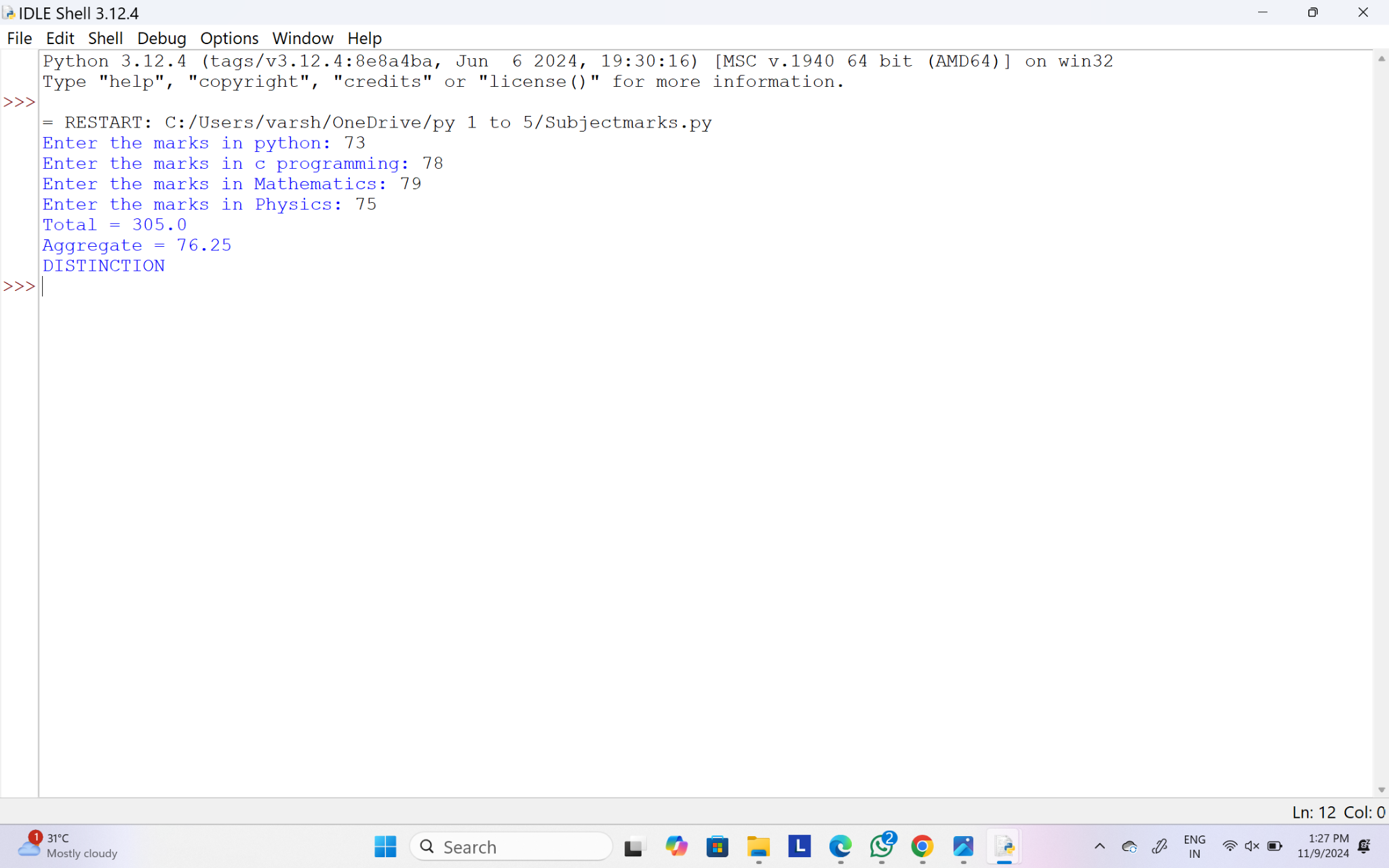
if \_\_name\_\_ == "\_\_main\_\_":

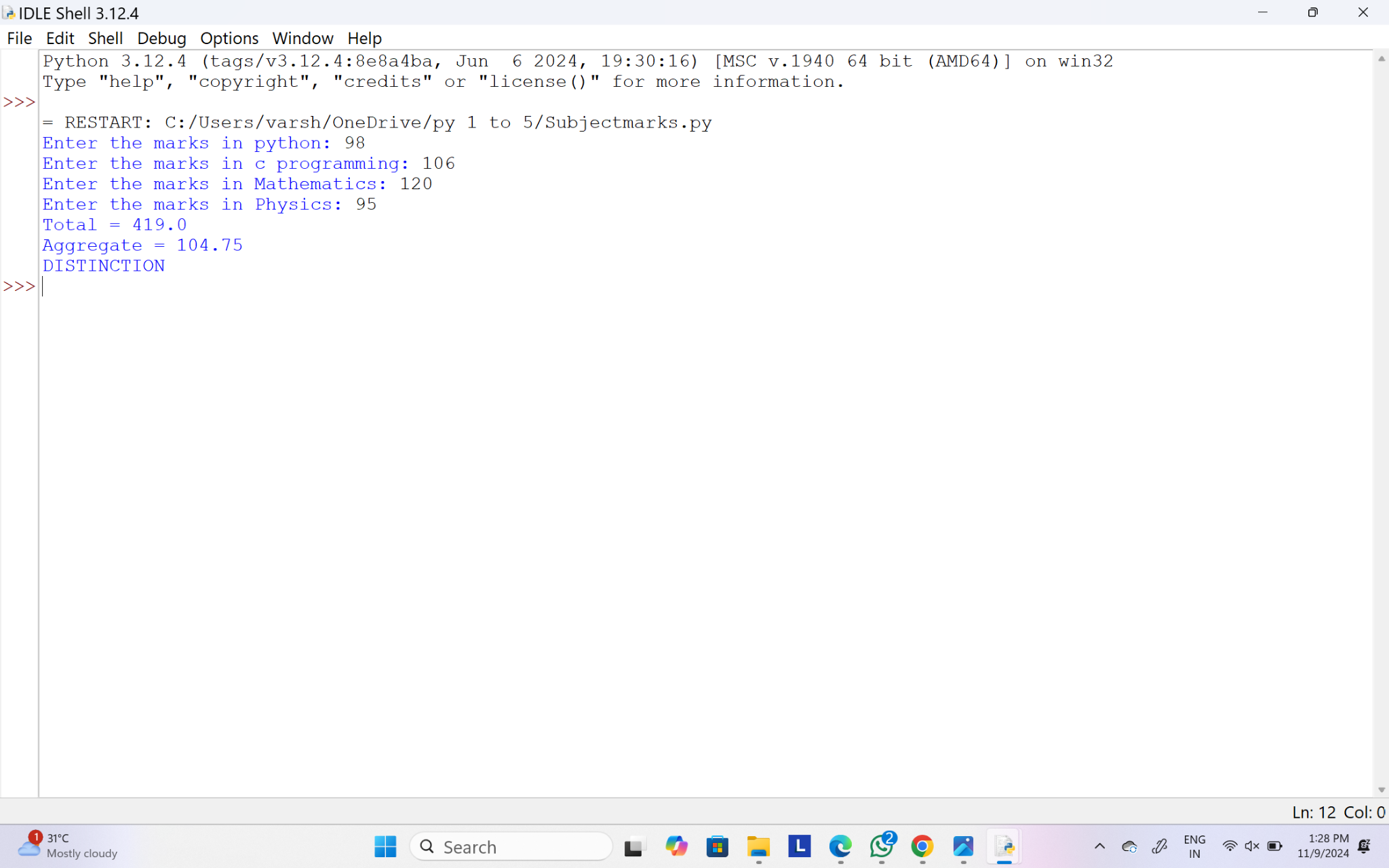
main()

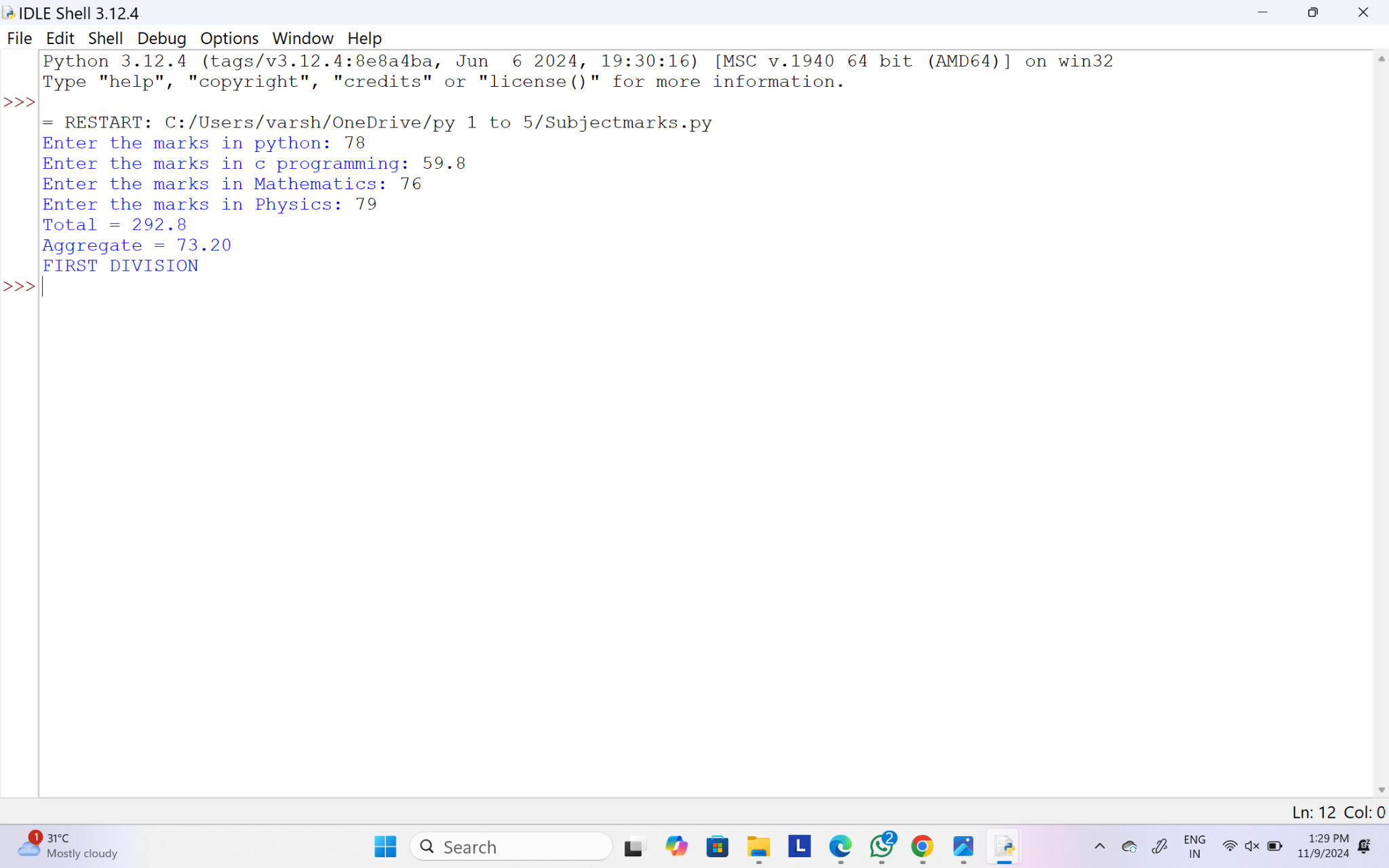
**OUTPUT:**

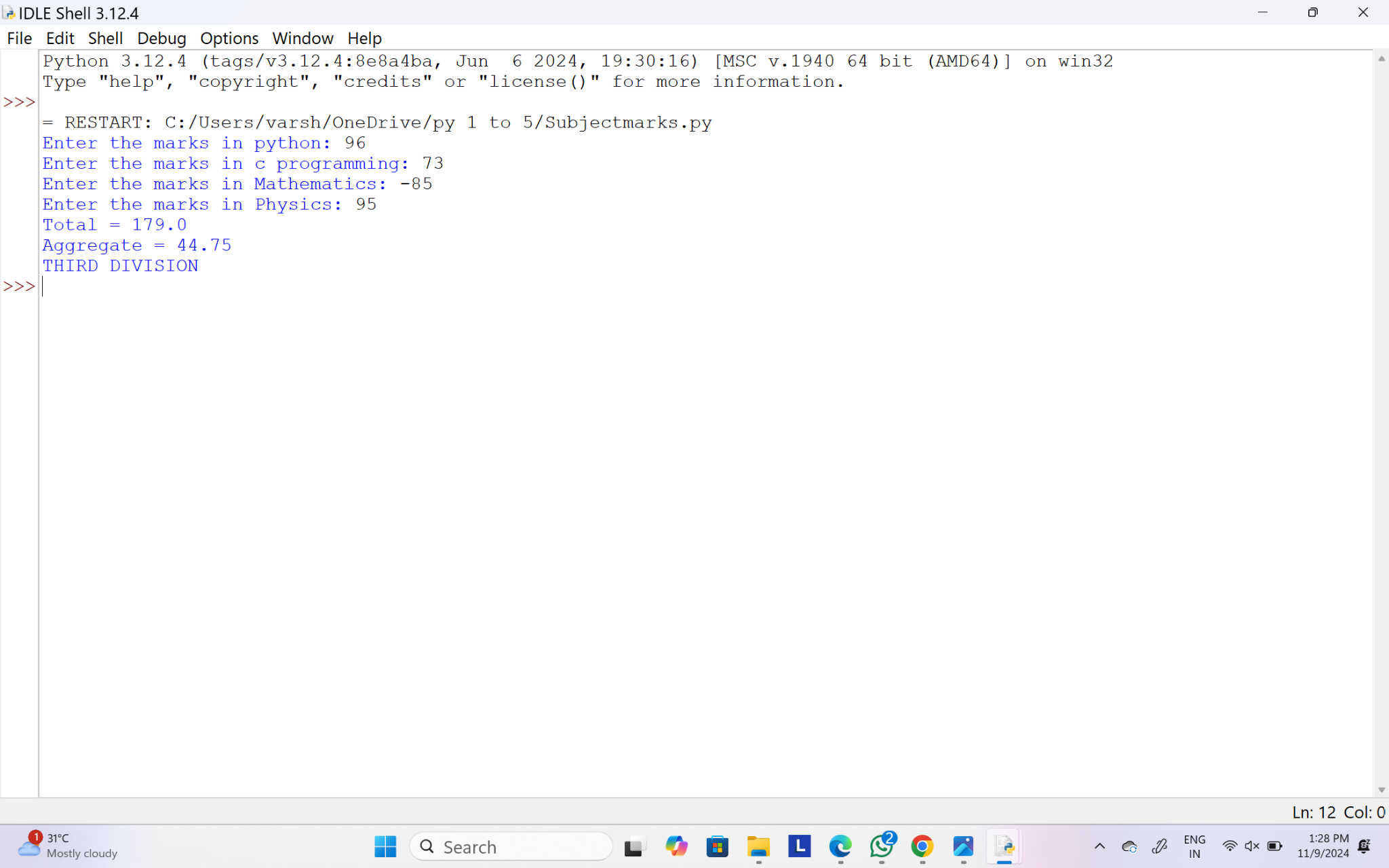
****

****

****

****

****

****

23.Write a program to print the numbers from M to N by skipping K numbers in between?

Sample Input:

M = 50

N = 100

K = 7

Sample Output:

50, 58, 65, 72, …..

**Test cases:**

1. M = 15, N = 05, K = 02
2. .M = 25, N = 50, K = 04
3. M = 15, N = 100, K = -02
4. M = 0 , N = 0 , K = 2
5. M = 200 , N = 200 , K = 50

**PROGRAM:**

def print\_numbers(M, N, K):

if K > 0:

if M > N:

print("Invalid input: M cannot be greater than N when K is positive.")

else:

numbers = range(M, N + 1, K + 1) # K+1 for proper skipping

print(", ".join(map(str, numbers)))

elif K < 0:

if M < N:

print("Invalid input: M cannot be less than N when K is negative.")

else:

numbers = range(M, N - 1, K - 1) # K-1 for proper skipping

print(", ".join(map(str, numbers)))

elif K == 0:

print("Invalid input: K cannot be zero.")

def run\_tests():

test\_cases = [

(50, 100, 7),

(15, 5, 2),

(25, 50, 4),

(15, 100, -2),

(0, 0, 2),

(200, 200, 50)

]

for M, N, K in test\_cases:

print(f"Input: M = {M}, N = {N}, K = {K}")

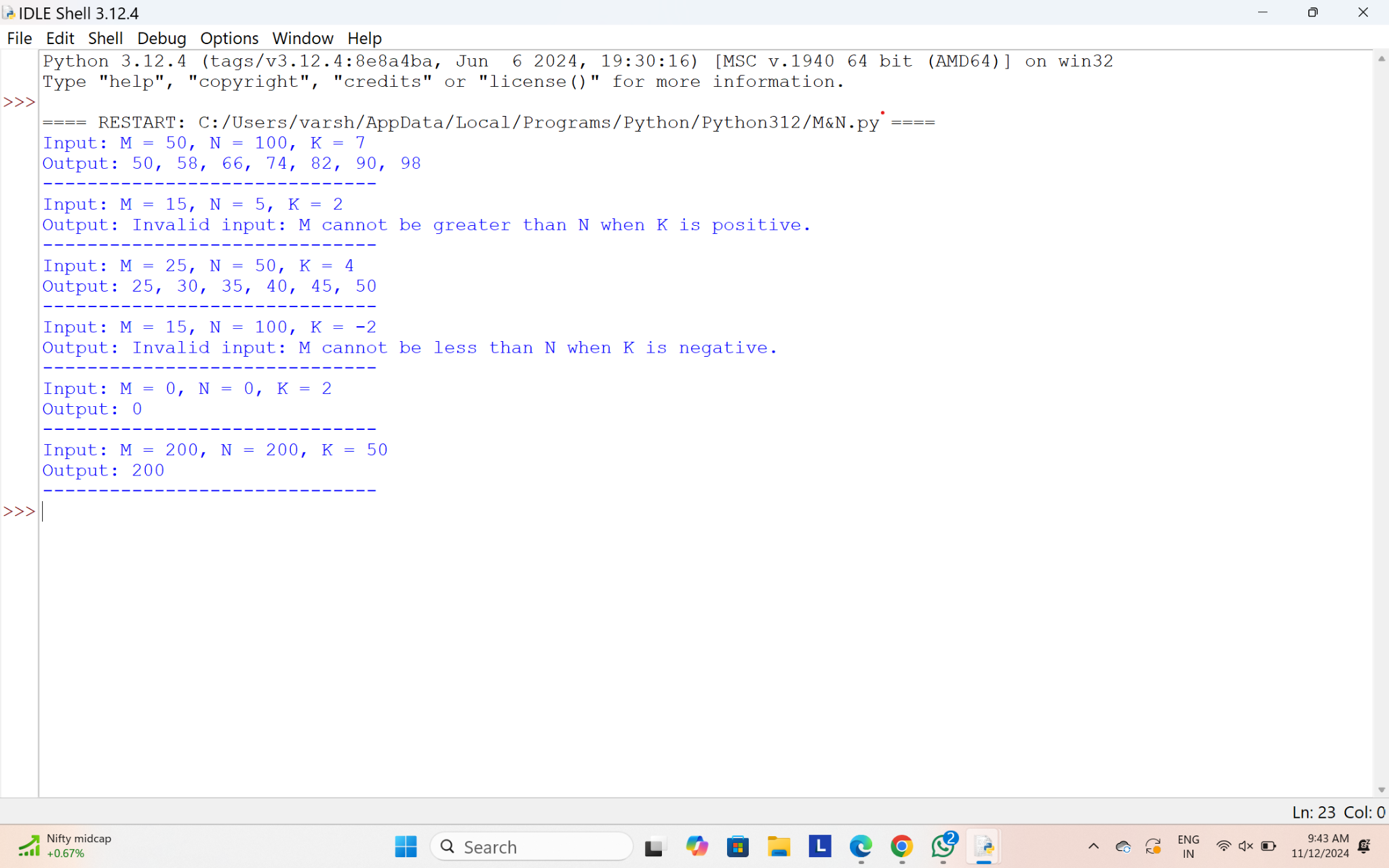
print("Output:", end=" ")

print\_numbers(M, N, K)

print("-" \* 30)

run\_tests()

**OUTPUT:**

****

24.Write a program for matrix addition?

Sample Input:

Mat1 = 1 2

5 3

Mat2 = 2 3

4 1

Sample Output:

Mat Sum = 3 5

9 4

**PROGRAM:**

def add\_matrices(mat1, mat2):

if len(mat1) != len(mat2) or len(mat1[0]) != len(mat2[0]):

print("Error: Matrices must have the same dimensions to be added.")

return None

result = []

for i in range(len(mat1)):

row = []

for j in range(len(mat1[0])):

row.append(mat1[i][j] + mat2[i][j])

result.append(row)

return result

def print\_matrix(matrix):

for row in matrix:

print(" ".join(map(str, row)))

mat1 = [

[1, 2],

[5, 3]

]

mat2 = [

[2, 3],

[4, 1]

]

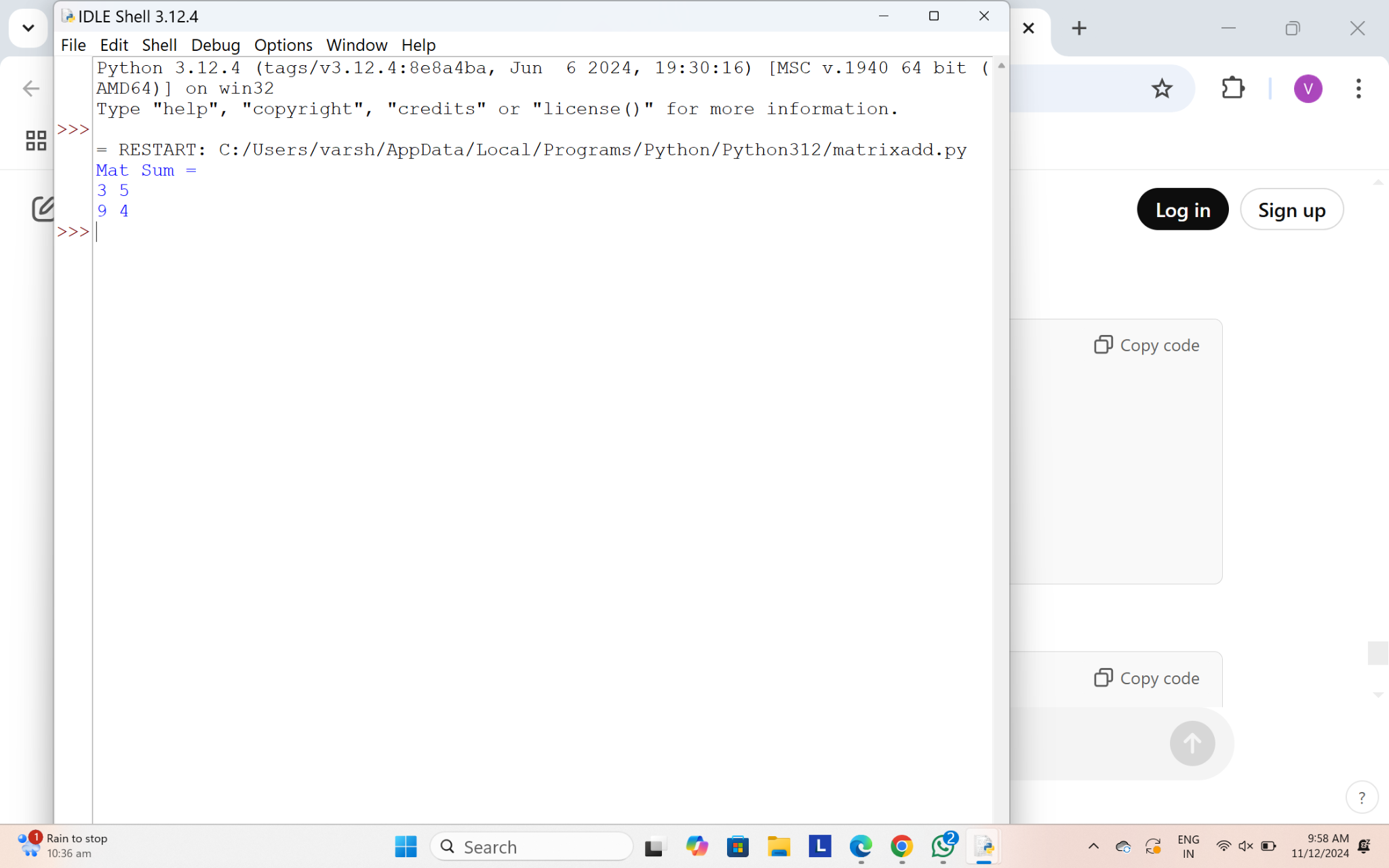
mat\_sum = add\_matrices(mat1, mat2)

if mat\_sum:

print("Mat Sum =")

print\_matrix(mat\_sum)

**OUTPUT:**

****

25.Write a program to calculate tax given the following conditions:

* 1. If income is less than or equal to 1,50,000 then no tax
  2. If taxable income is 1,50,001 – 3,00,000 the charge 10% tax
  3. If taxable income is 3,00,001 – 5,00,000 the charge 20% tax
  4. If taxable income is above 5,00,001 then charge 30% tax

Sample Input:

Enter the income:200000

Sample Output:

Tax= 20000

**Test cases:**

1. 400700
2. 2789239
3. 150000
4. 00000
5. -125486

**PROGRAM:**

def calculate\_tax(income):

if income < 0:

return "Invalid income"

if income <= 150000:

return 0

elif income <= 300000:

return 0.10 \* income

elif income <= 500000:

return 0.20 \* income

else:

return 0.30 \* income

income = float(input("Enter the income: "))

tax = calculate\_tax(income)

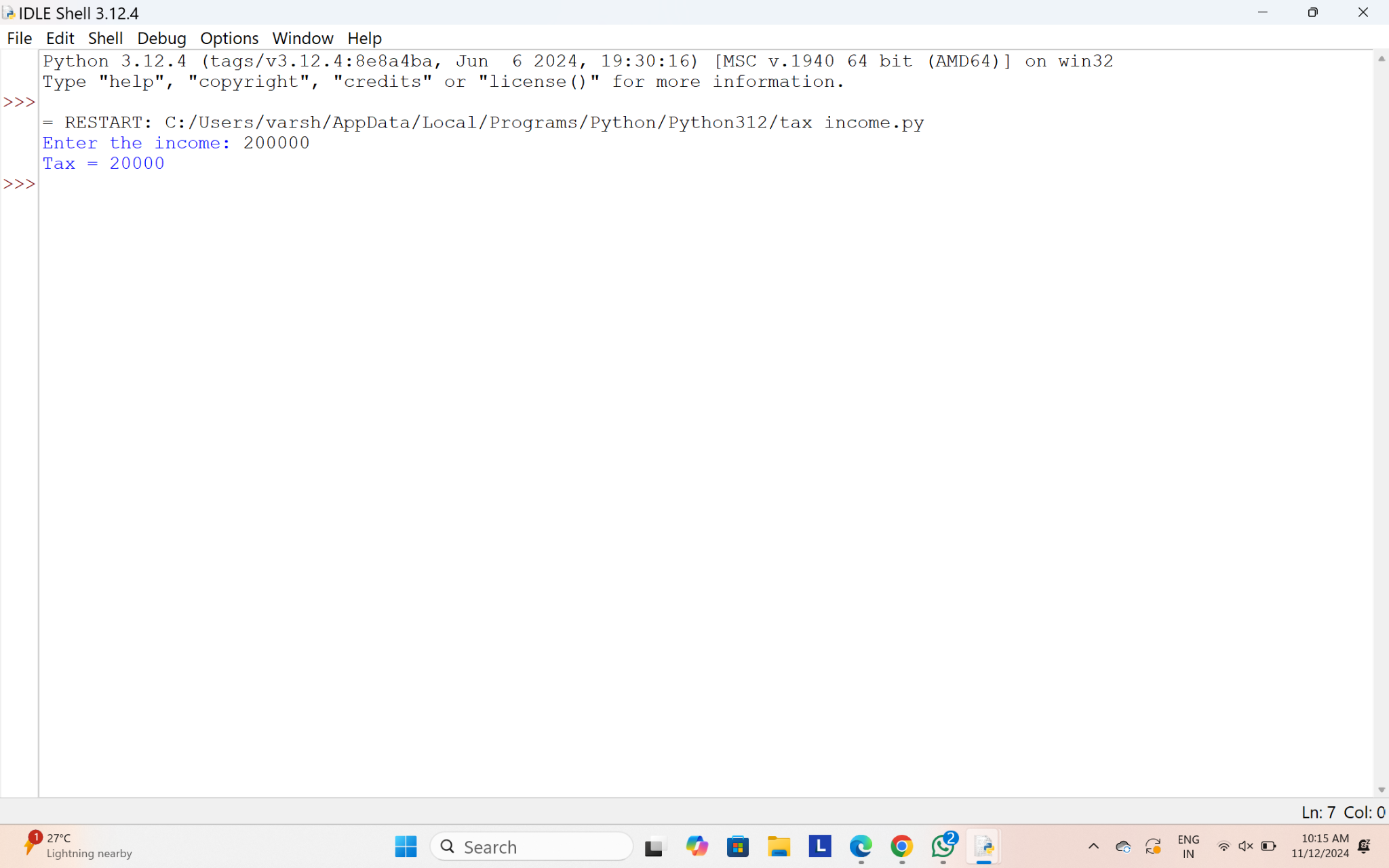
if isinstance(tax, str):

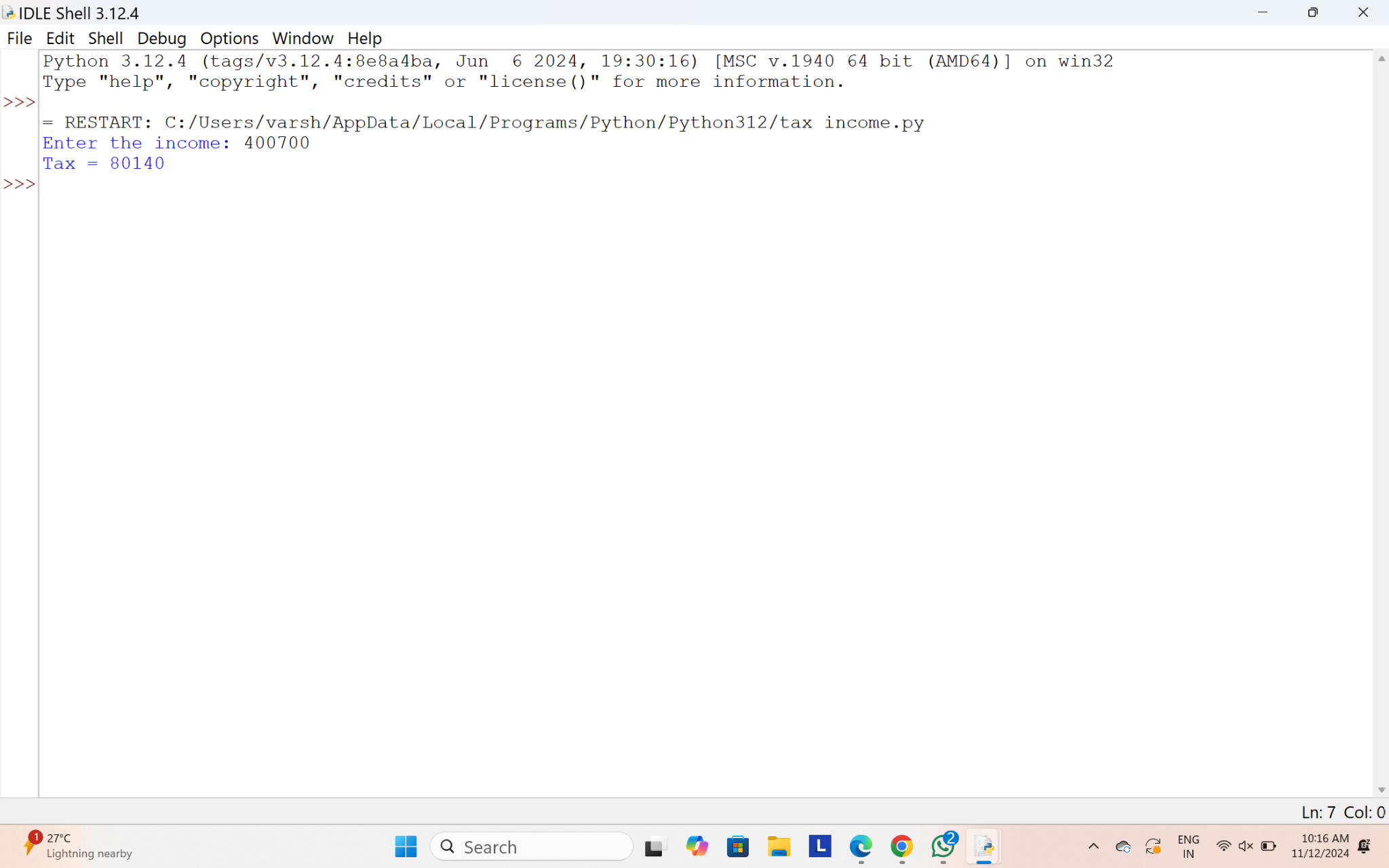
print(tax) # In case of invalid income

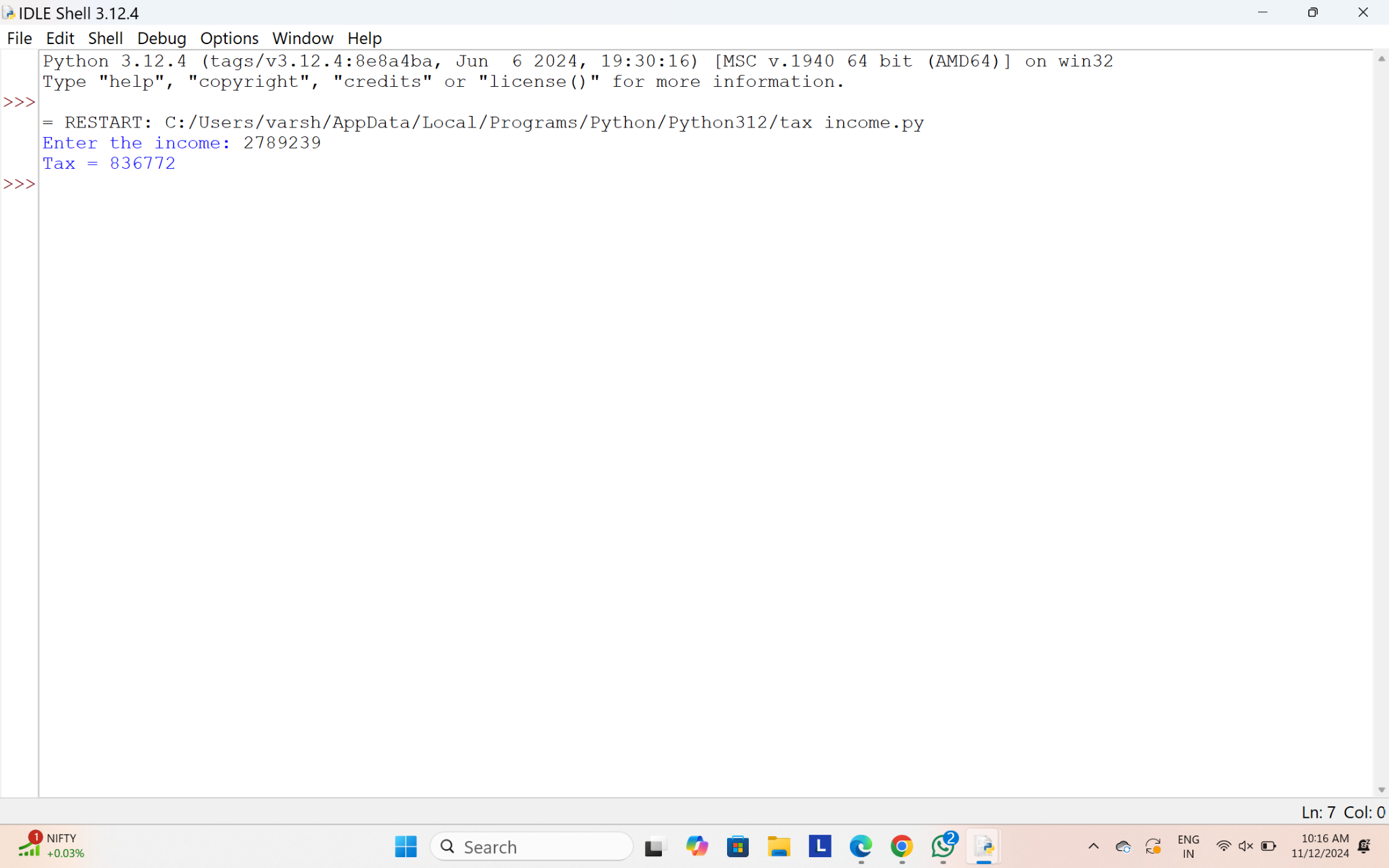
else:

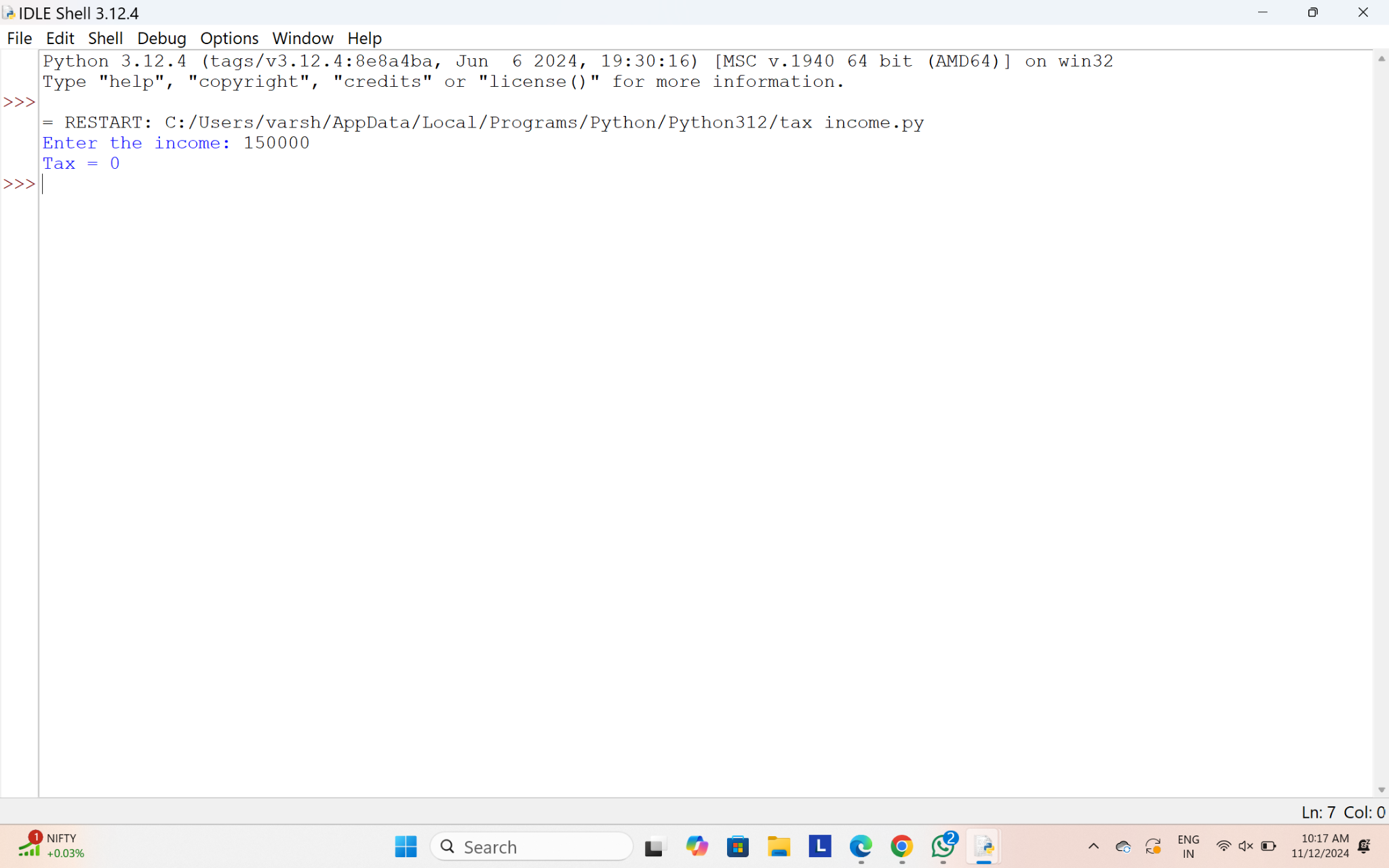
print(f"Tax = {tax:.0f}")

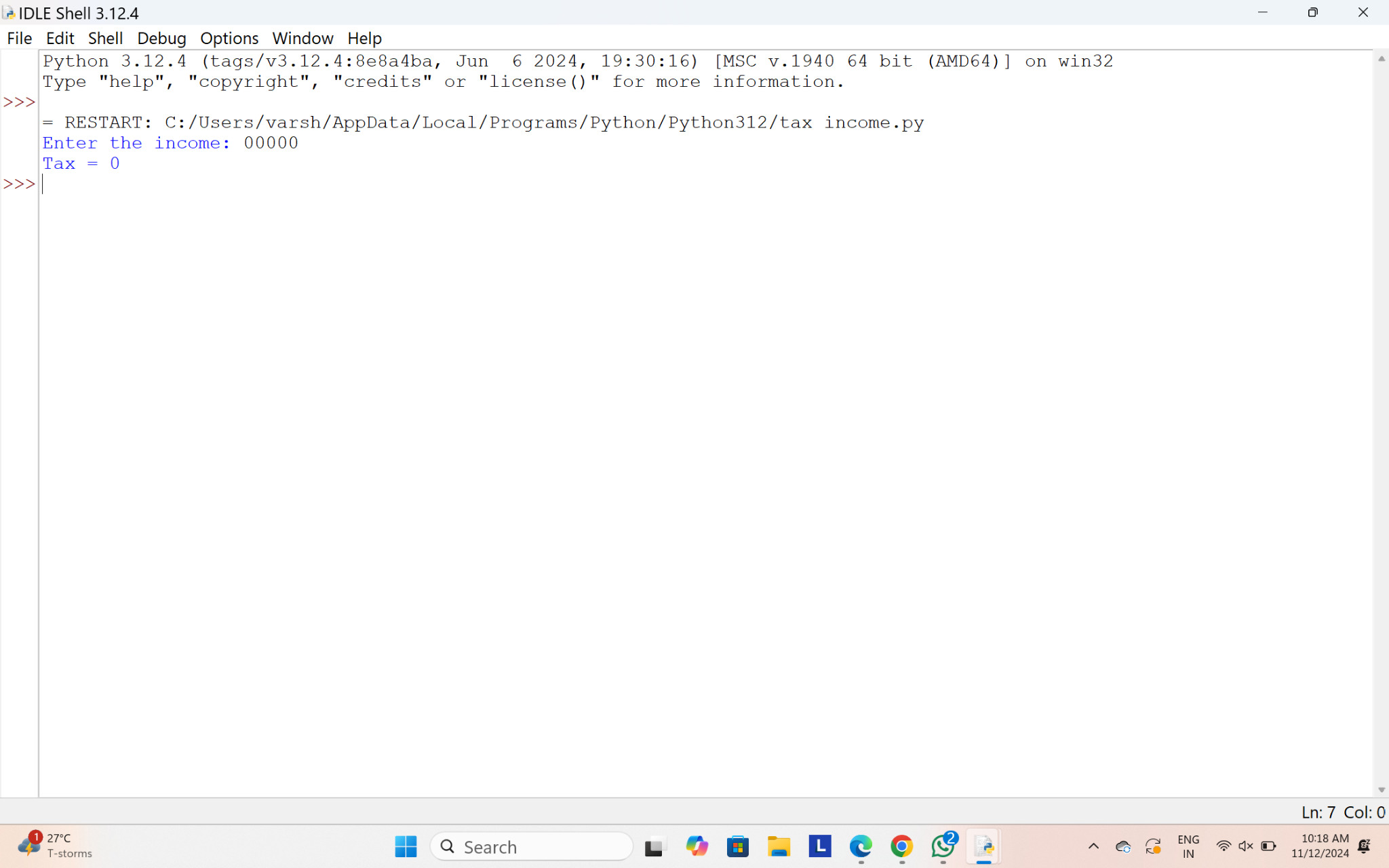
**OUTPUT:**

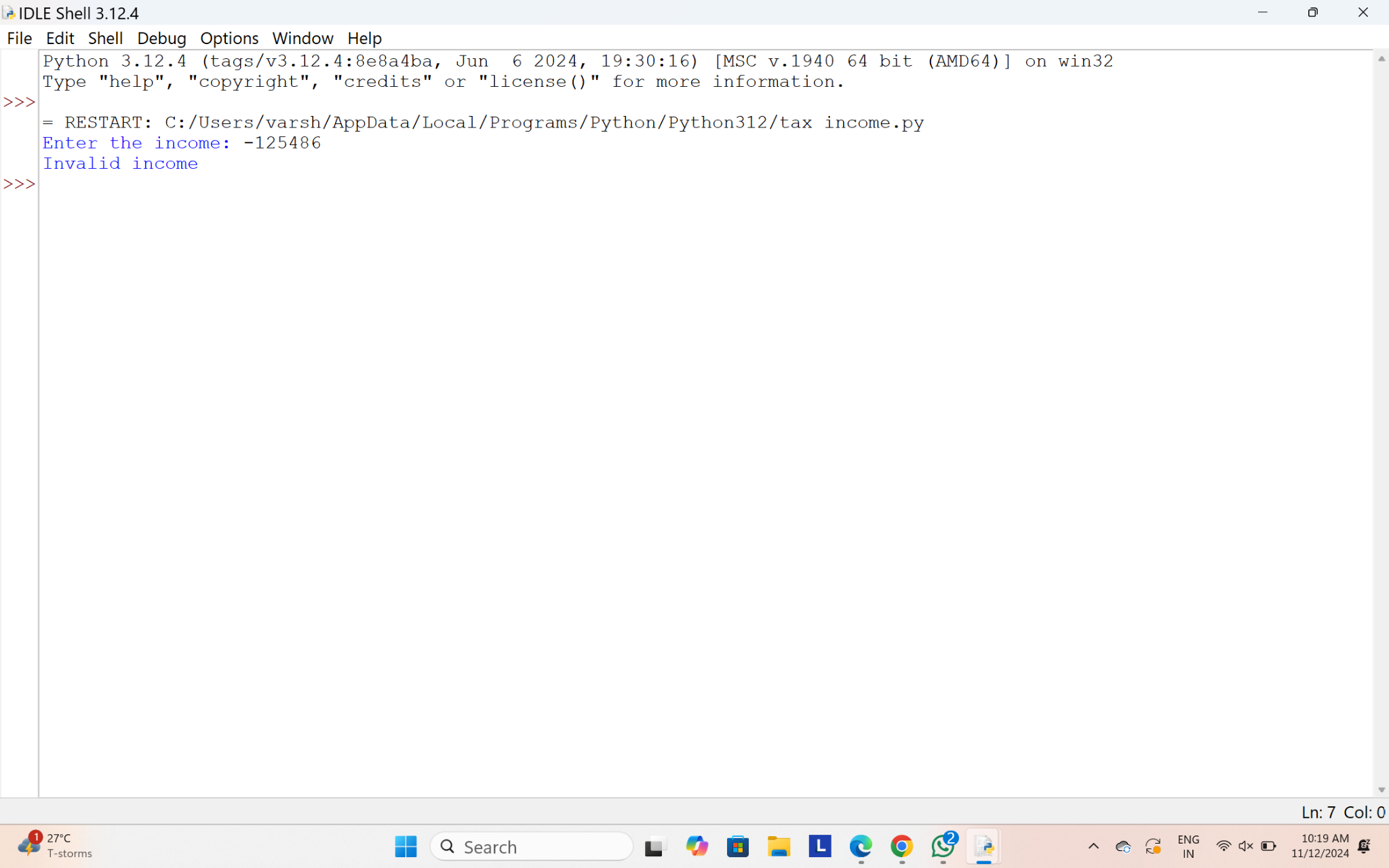
****

****

****

****

****

****

26.Write a program that would sort a list of names in alphabetical order Ascending or Descending, choice get from the user?

Sample Input:

Banana

Carrot

Radish

Apple

Jack

Order(A/D) : A

Sample Output:

Apple

Banana

Carrot

Jack

Radish

**PROGRAM:**

def sort\_names(names, order):

if order == 'A':

names.sort()

elif order == 'D':

names.sort(reverse=True)

return names

names = []

print("Enter names (type 'done' to finish):")

while True:

name = input()

if name.lower() == 'done':

break

names.append(name)

order = input("Order(A/D): ").upper()

if order not in ['A', 'D']:

print("Invalid order choice!")

else:

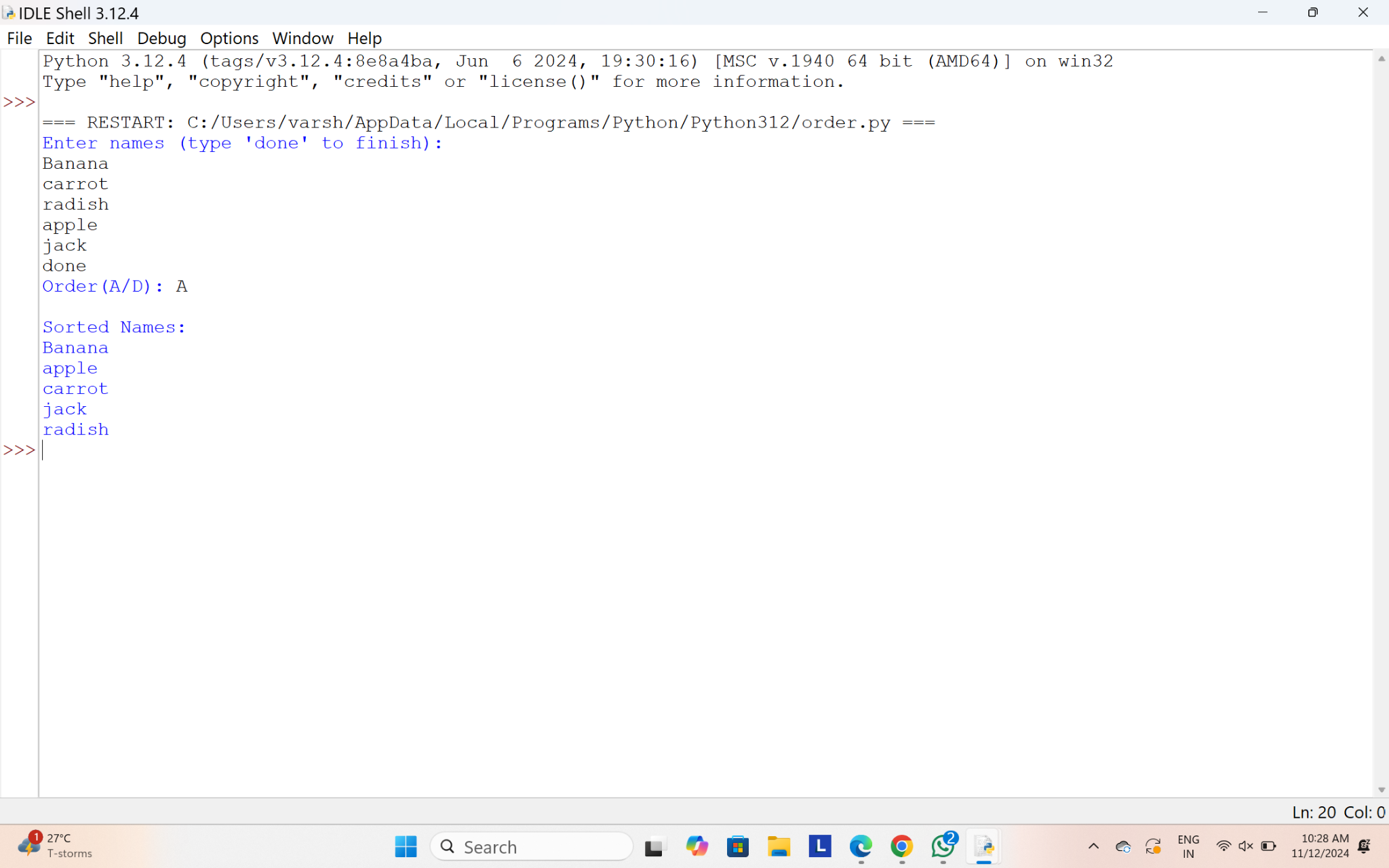
sorted\_names = sort\_names(names, order)

print("\nSorted Names:")

for name in sorted\_names:

print(name)

**OUTPUT:**

****

27.Write a program for matrix multiplication?

Sample Input:

Mat1 = 1 2

5 3

Mat2 = 2 3

4 1

Sample Output:

Mat Sum = 10 5

22 18

**PROGRAM:**

def matrix\_multiply(mat1, mat2):

rows\_mat1 = len(mat1)

cols\_mat1 = len(mat1[0])

rows\_mat2 = len(mat2)

cols\_mat2 = len(mat2[0])

if cols\_mat1 != rows\_mat2:

return "Matrix multiplication is not possible. Invalid dimensions."

result = [[0 for \_ in range(cols\_mat2)] for \_ in range(rows\_mat1)]

for i in range(rows\_mat1):

for j in range(cols\_mat2):

for k in range(cols\_mat1): # or equivalently: for k in range(rows\_mat2)

result[i][j] += mat1[i][k] \* mat2[k][j]

return result

mat1 = [

[1, 2],

[5, 3]

]

mat2 = [

[2, 3],

[4, 1]

mat\_sum = matrix\_multiply(mat1, mat2)

if isinstance(mat\_sum, str):

print(mat\_sum)

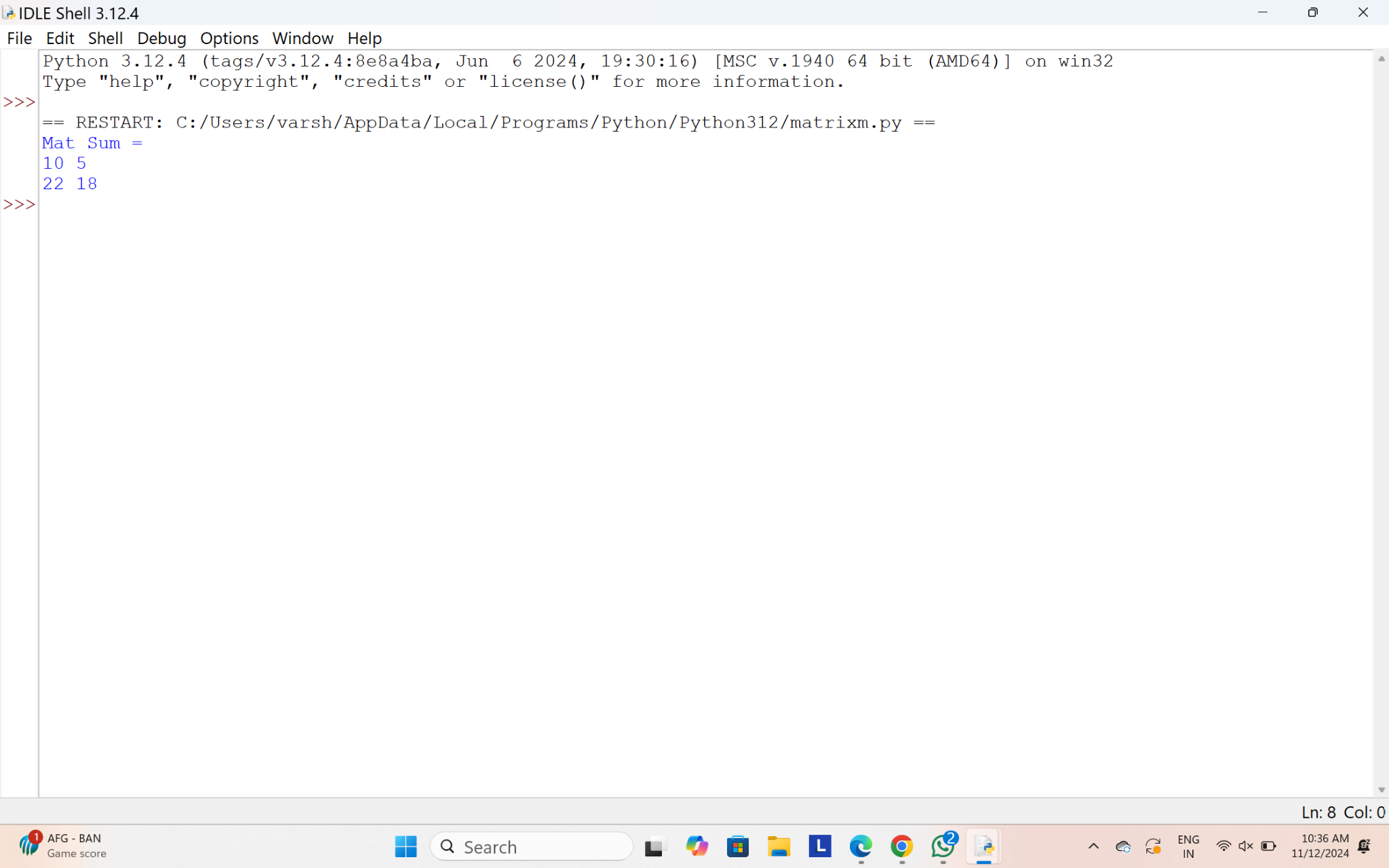
else:

print("Mat Sum = ")

for row in mat\_sum:

print(" ".join(map(str, row)))

**OUTPUT:**

****

28.and aggregate, display the grade obtained by the student. If the student scores an aggregate greater than 75%, then the grade is Distinction. If aggregate is 60>= and <75, then the grade is First Division. If aggregate is 50 >= and <60, then the grade is Second Division. If aggregate is 40>= and <50, then the grade is Third Division. Else the grade is Fail.

Sample Input & Output:

Enter the marks in python: 90

Enter the marks in c programming: 91

Enter the marks in Mathematics: 92

Enter the marks in Physics: 93

Total= 366

Aggregate = 91.5

DISTINCTION

**Test cases:**

1. 18, 76,93,65
2. 73,78,79,75
3. 98,106,120,95
4. 96,73, -85,95
5. 78,59.8,76,79

**PROGRAM:**

def calculate\_grade(marks):

total\_marks = sum(marks)

aggregate = total\_marks / len(marks)

print(f"Total = {total\_marks}")

print(f"Aggregate = {aggregate}")

if aggregate >= 75:

print("DISTINCTION")

elif 60 <= aggregate < 75:

print("FIRST DIVISION")

elif 50 <= aggregate < 60:

print("SECOND DIVISION")

elif 40 <= aggregate < 50:

print("THIRD DIVISION")

else:

print("FAIL")

def main():

test\_cases = [

[18, 76, 93, 65],

[73, 78, 79, 75],

[98, 106, 120, 95],

[96, 73, -85, 95],

[78, 59.8, 76, 79]

]

for marks in test\_cases:

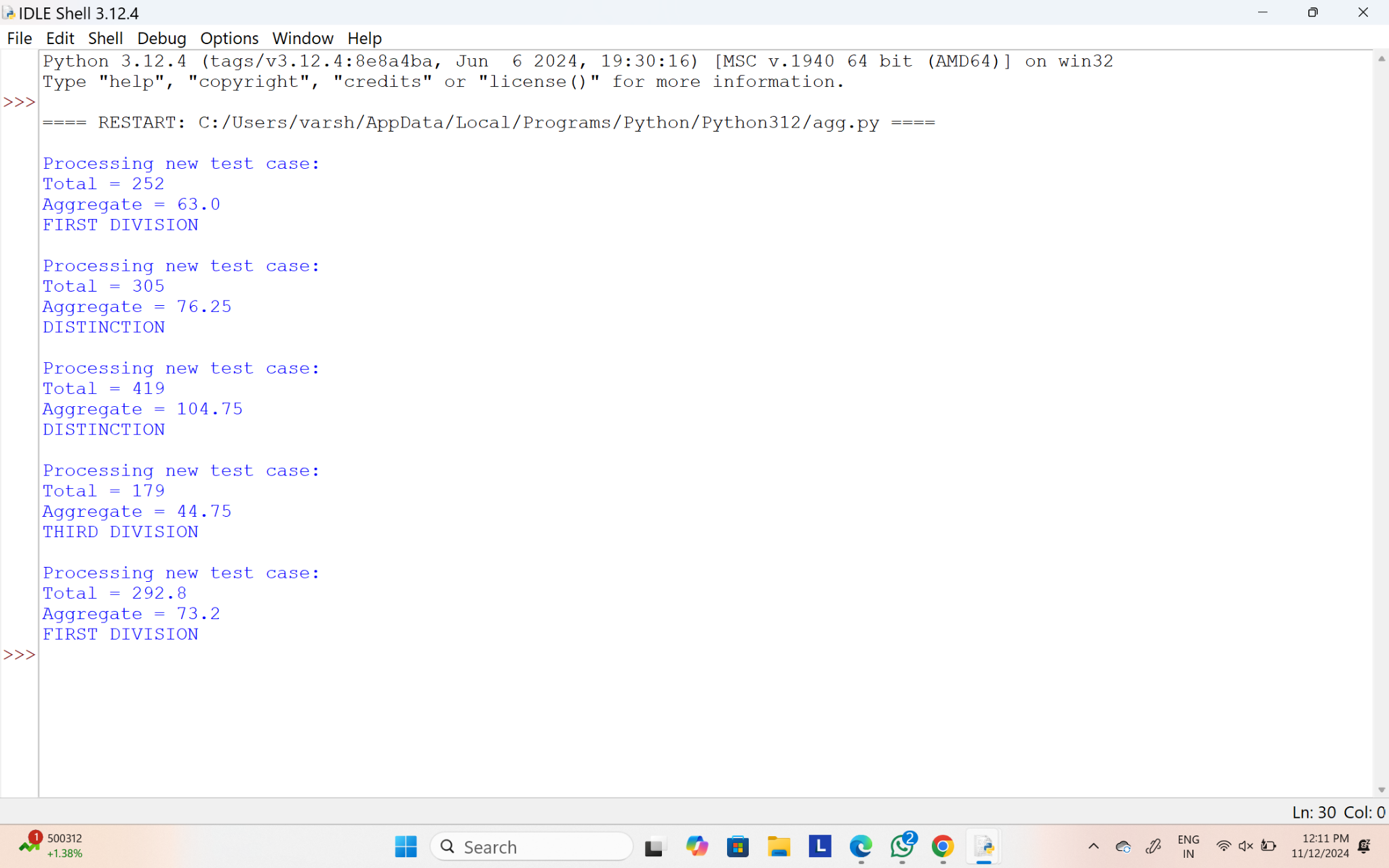
print("\nProcessing new test case:")

calculate\_grade(marks)

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

main()

**OUTPUT:**

****

29. Write a program to print the following pattern

Sample Input:

Enter the number to be printed: 1

Max Number of time printed: 3

1

11

111

11

1

**PROGRAM:**

def print\_pattern(n, m):

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

print(str(n) \* i)

for i in range(m - 1, 0, -1):

print(str(n) \* i)

def main():

n = int(input("Enter the number to be printed: "))

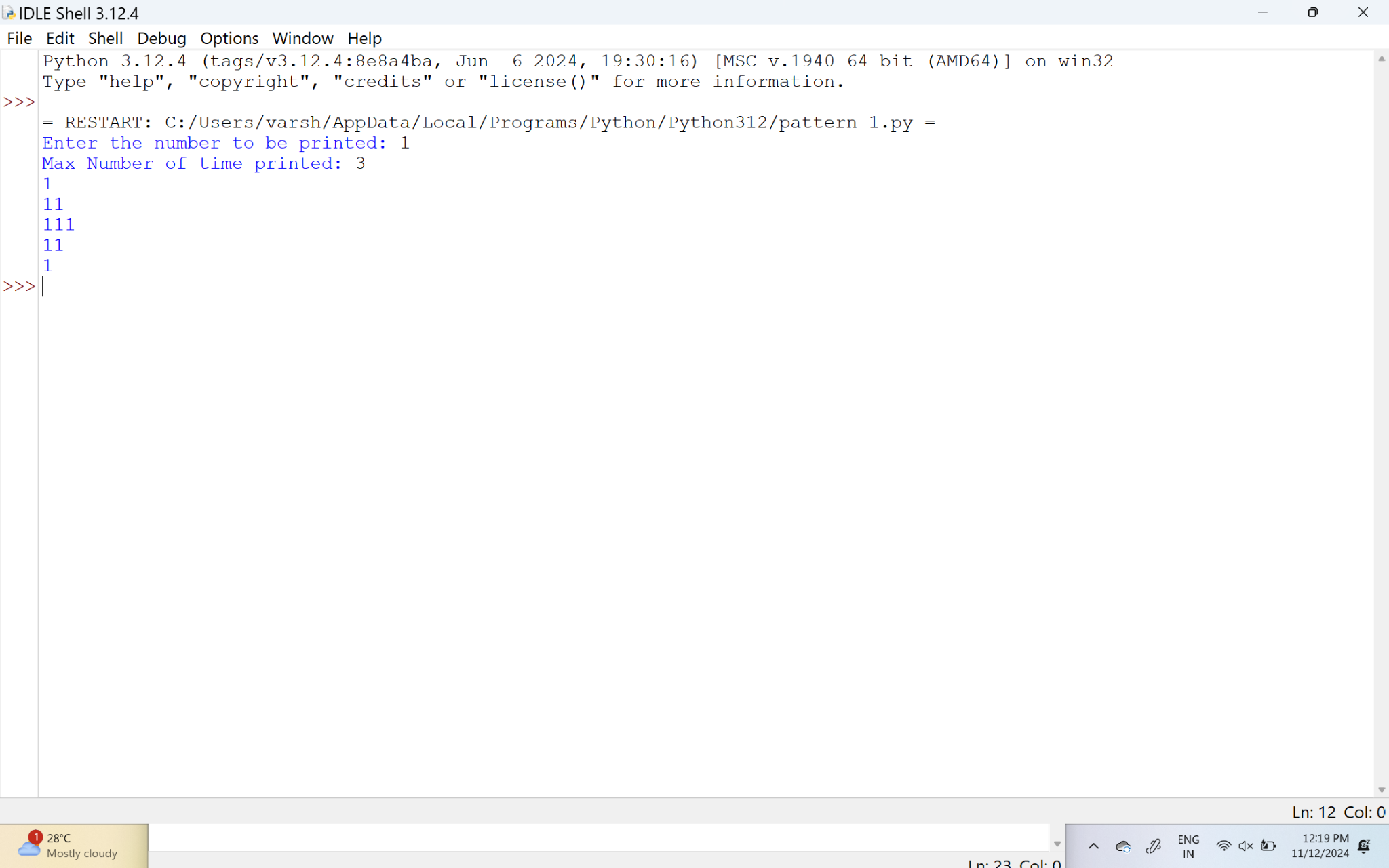
m = int(input("Max Number of time printed: "))

print\_pattern(n, m)

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

main()

**OUTPUT:**

****

30.Write a program to print the special characters separately and print number of Special characters in the line?

**PROGRAM:**

import string

def print\_special\_characters(input\_string):

special\_char\_count = 0

special\_chars = []

for char in input\_string:

if char not in string.ascii\_letters + string.digits + ' ':

special\_chars.append(char)

special\_char\_count += 1

for char in special\_chars:

print(char)

print(f"Number of Special Characters: {special\_char\_count}")

def main():

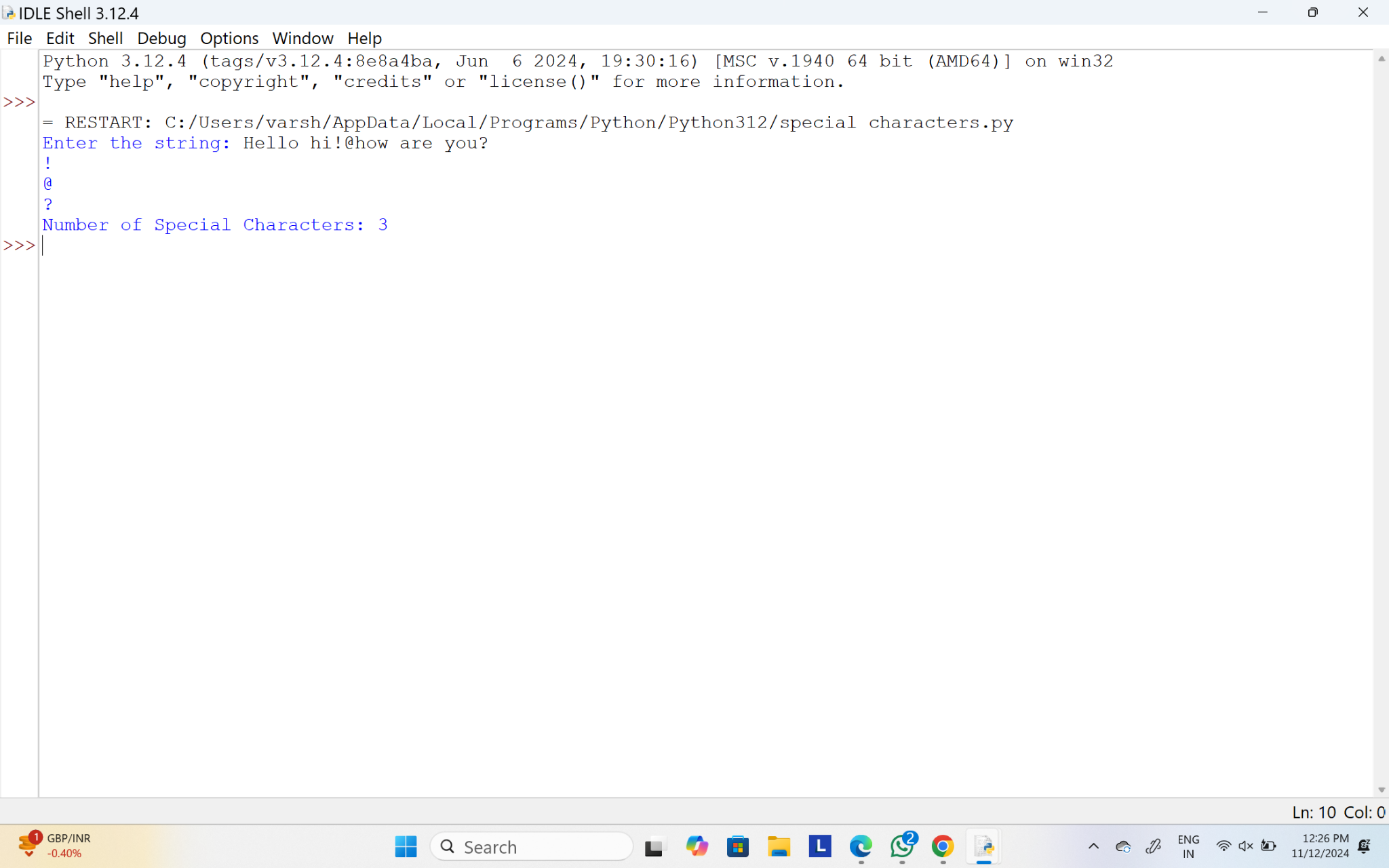
input\_string = input("Enter the string: ")

print\_special\_characters(input\_string)

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

main()

**OUTPUT:**

****