**PYTHON PROGRAMMING**

**LAB MANUAL**

Prepared by

P.SASANK REDDY

DEPT.OF COMPUTER ENGINEERING

2ND SHIFT POLYTECHNIC BAPATLA ENGINEERING COLLEGE

BAPATLA

|  |  |
| --- | --- |
| **SI.**  **NO** | **EXPERIMENTS NAME** |
| 1. | Write and execute simple python Program. |
| 2. | Write/execute simple ‘python’ program: Develop minimum 2 programs using different data types  (numbers, string, tuple, list, dictionary). |
| 3. | Write/execute simple ‘Python’ program: Develop minimum 2 programs using Arithmetic Operators,  Exhibiting data type conversion. |
| 4. | i) Write simple programs to convert U.S dollars to indian rupees.  ii) Write simple programs to convert bits to Megabyte, Gigabytes and Terabytes. |
| 5. | Write simple programs to calculate the area and perimeter of the square, and the volume & perimeter of the cone. |
| 6. | Write program to: (i) Determine whether a given number is odd or even .  (ii). Find the greatest of three numbers using conditional operator. |
| 7. | Write a program to : (i) Find factorial of a given number.  (ii) Generate multiplication table up to 10 for numbers 1 to 5. |
| 8. | Write a program to: (i) Find factorial of a given number using functions.  (ii) Generate multiplication table up to 10 for numbers 1 to 5 using functions. |
| 9. | Write a program to: (i) Find factorial of a given number using recursion.  (ii)Generate Fibonacci sequence up to 100 using recursion. |
| 10. | Write a program to: To print Factors of a given Number. |
| 11. | Write a program to: Create a list, add elements to list, delete elements from the list. |
| 12. | Write a program to: Sort the list, reverse the list and counting elements in a list. |
| 13. | Write a program to: Create dictionary, add elements to dictionary, delete element from the dictionary. |
| 14. | Write a program to: To calculate average, mean, median, and standard deviations of numbers in a  List |
| 15. | File input/output: Write a program to: (i) To create simple file and write “Hello World” in it.  (ii)To open a file in write mode and append Hello world at the end of a file. |
| 16. | Write a program to: (i) To open a file but replace every occurrence of character ‘h’  (ii) To open a file in read mode and print the number of occurrences of a character ‘a’ |
| 17. | Write a program to: Add two complex number using classes and objects. |
| 18. | Write a program to: Subtract two complex numbers using classes and objects. |
| 19. | Write a program to : Create a package and accessing a package. |

**1. Aim :** Write and execute simple Python program.

**Source Code :**

*# Read the student details*

name = input("Enter the student name: ")

department = input("Enter the student department: ")

academic\_year = input("Enter the student academic year: ")

pin\_number = input("Enter the student PIN number: ")

*# Print the student details*

*# Using the f-string*

print("\nStudent Details:")

print(f"Name: {name}")

print(f"Department: {department}")

print(f"Academic Year: {academic\_year}")

print(f"PIN Number: {pin\_number}")

*#print(f"\nStudent Details:\nName: {name}\nDepartment: {department}\nAcademic Year: {academic\_year}\nPIN Number: {pin\_number}")*

*"""*

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

*#Other ways to print format in Python*

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

*# Using the format() method*

*print("Name: {}".format(name))*

*print("Department: {}".format(department))*

*print("Academic Year: {}".format(academic\_year))*

*print("PIN Number: {}".format(pin\_number))*

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

*# Using the % formatting*

*print("Name: %s" % name)*

*print("Department: %s" % department)*

*print("Academic Year: %s" % academic\_year)*

*print("PIN Number: %s" % pin\_number)*

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

*# Using the string concatenation*

*print("Name: " + name)*

*print("Department: " + department)*

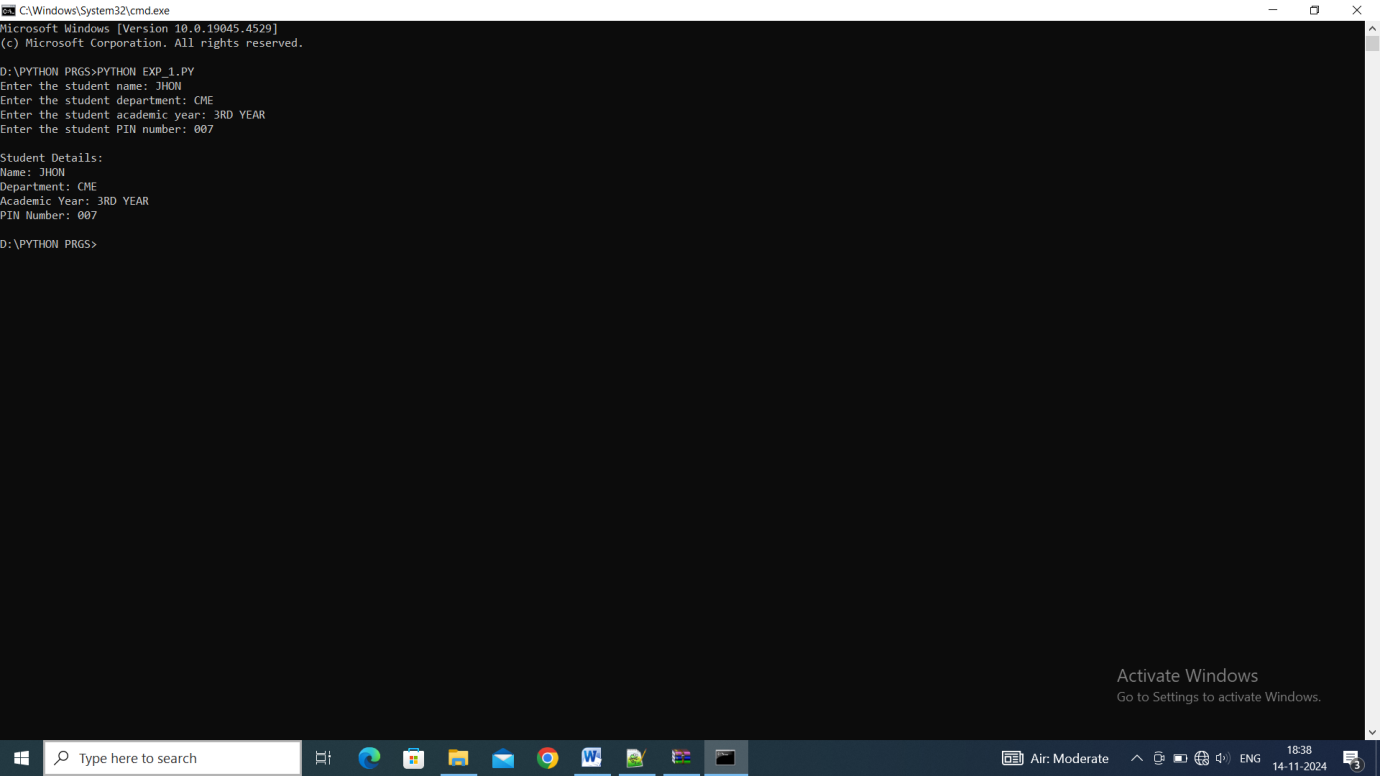
*print("Academic Year: " + academic\_year)*

*print("PIN Number: " + pin\_number)*

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

*"""*

**OUTPUT:**

**

**2.A. Aim :** Write a Python Program to perform relational and logical operations on two integer inputs and display the results, demonstrating basic comparison and conditional logic in Python.

**Source Code :**

*# Take two integer inputs from the user*

num1 = int(input("Enter the first number: "))

num2 = int(input("Enter the second number: "))

*# Perform relational operations*

print(f"\nRelational Operations between {num1} and {num2}:")

print(f"{num1} > {num2}  is {num1 > num2}")

print(f"{num1} < {num2}  is {num1 < num2}")

print(f"{num1} == {num2}  is {num1 == num2}")

print(f"{num1} != {num2}  is {num1 != num2}")

print(f"{num1} >= {num2}  is {num1 >= num2}")

print(f"{num1} <= {num2}  is {num1 <= num2}")

*# Perform logical operations*

print(f"\nLogical Operations between {num1} and {num2}:")

is\_positive1 = num1 > 0

is\_positive2 = num2 > 0

print(f"Both numbers are positive: {is\_positive1 and is\_positive2}")

print(f"At least one number is positive: {is\_positive1 or is\_positive2}")

print(f"First number is not positive: {not is\_positive1}")

print(f"Second number is not positive: {not is\_positive2}")

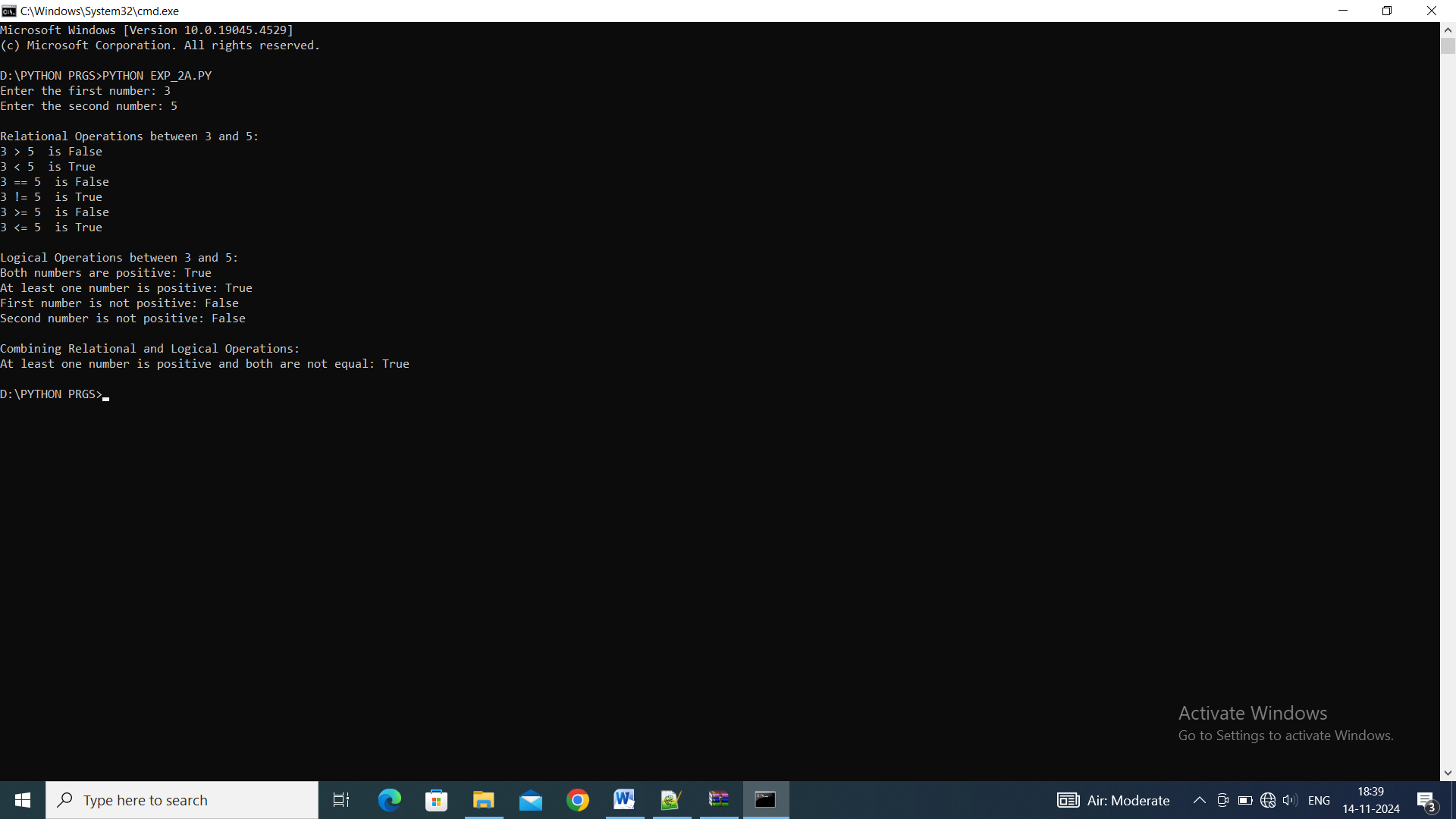
*# Combining relational and logical operations*

print(f"\nCombining Relational and Logical Operations:")

X = (is\_positive1 or is\_positive2) and (num1 != num2)

print(f"At least one number is positive and both are not equal: {X}")

**OUTPUT:**

****

**2.B.Aim :** Write a Python Program to demonstrate various string manipulations in Python, including measuring string length, changing case, replacing substrings, checking for substring presence, concatenating strings, and slicing.

**Source Code :**

*# Get string input from the user*

user\_string = input("\nEnter a string: ")

print(f"You entered: {user\_string}")

*# String length*

string\_length = len(user\_string)

print(f"Length of the string: {string\_length}")

*# Convert string to uppercase*

uppercase\_string = user\_string.upper()

print(f"Uppercase string: {uppercase\_string}")

*# Convert string to lowercase*

lowercase\_string = user\_string.lower()

print(f"Lowercase string: {lowercase\_string}")

*# Replace a substring*

old\_substring = input("\nEnter a substring to replace: ")

new\_substring = input("Enter the new substring: ")

user\_string = user\_string.replace(old\_substring, new\_substring)

print(f"String after replacement: {user\_string}")

*# Check if string contains a substring*

substring = input("\nEnter a substring to check: ")

contains\_substring = substring in user\_string

print(f"Does the string contain '{substring}'? {contains\_substring}")

*# String concatenation*

additional\_string = input("\nEnter another string to concatenate: ")

concatenated\_string = user\_string + " " + additional\_string

print(f"Concatenated string: {concatenated\_string}")

*# String slicing*

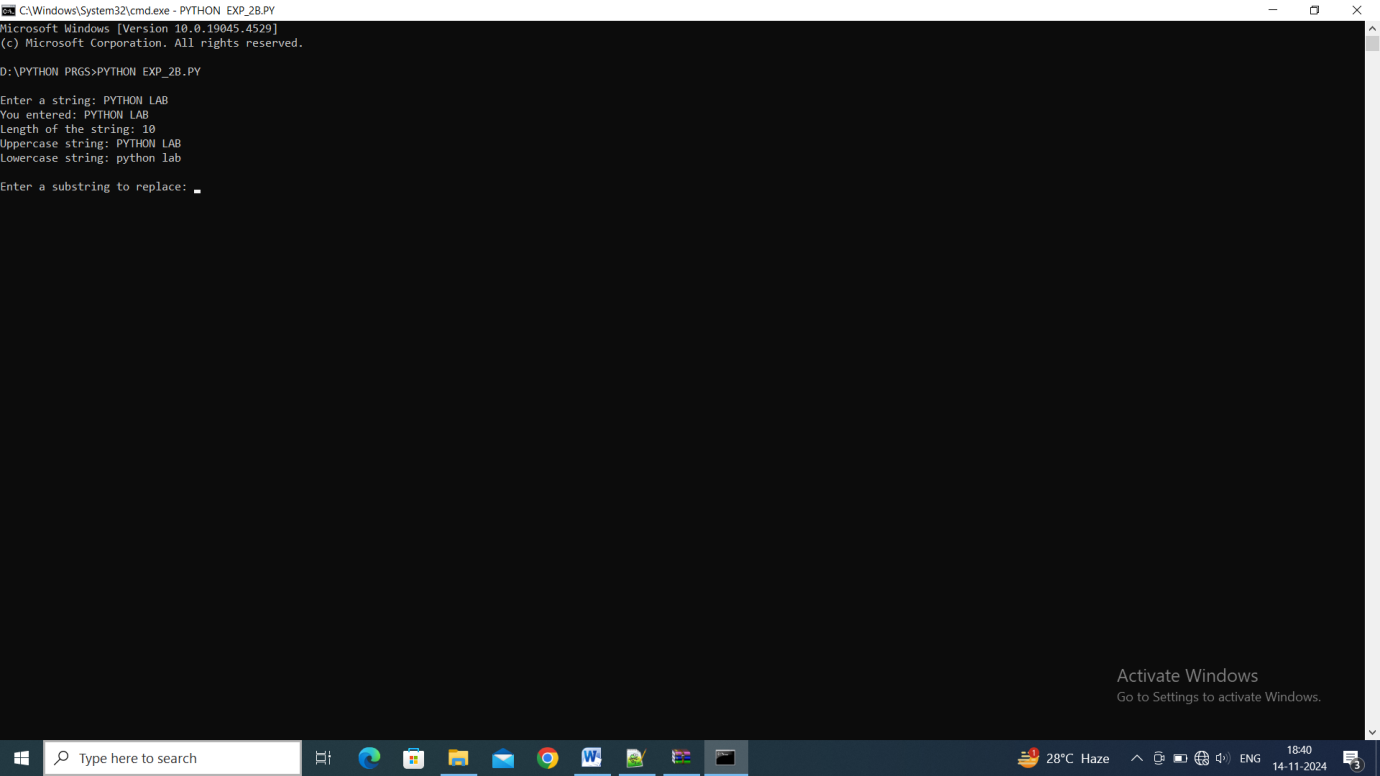
start\_index = int(input("\nEnter the start index for slicing: "))

end\_index = int(input("Enter the end index for slicing: "))

sliced\_string = user\_string[start\_index:end\_index]

print(f"Sliced string: {sliced\_string}")

**OUTPUT:**

****

**3.A. Aim :** Write a Python Program to perform and display basic arithmetic operations between two numbers.

**Source Code :**

*# Take two numbers as input from the user*

num1 = float(input("Enter the first number: "))

num2 = float(input("Enter the second number: "))

*# Perform arithmetic operations*

sum = num1 + num2

difference = num1 - num2

product = num1 \* num2

quotient = num1 / num2

remainder = num1 % num2

exponentiation = num1 \*\* num2

floor\_division = num1 // num2

*# Display the results*

print(f"\nArithmetic Operations between {num1} and {num2}:")

print(f"Sum: {num1} + {num2} = {sum}")

print(f"Difference: {num1} - {num2} = {difference}")

print(f"Product: {num1} \* {num2} = {product}")

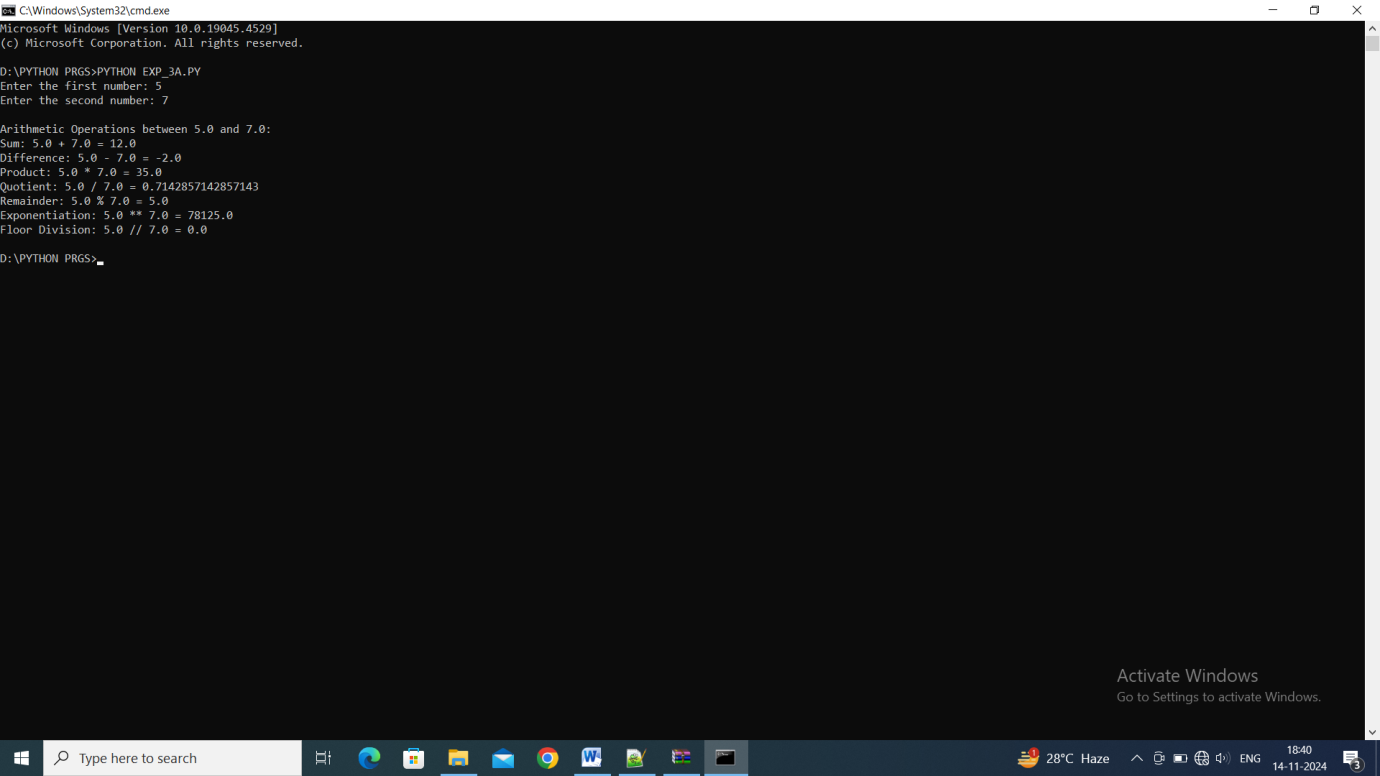
print(f"Quotient: {num1} / {num2} = {quotient}")

print(f"Remainder: {num1} % {num2} = {remainder}")

print(f"Exponentiation: {num1} \*\* {num2} = {exponentiation}")

print(f"Floor Division: {num1} // {num2} = {floor\_division}")

**OUTPUT:**

****

**3.B. Aim :** Write a Python Program to demonstrate data type conversions in Python with user input.

**Source Code :**

*# Input from the user*

num\_str = input("Enter a number: ")

print("\nData Type Conversion:")

*# Convert string to integer*

num\_int = int(num\_str)

print(f"String to Integer: {num\_str} -> {num\_int}")

*# Convert string to float*

num\_float = float(num\_str)

print(f"String to Float: {num\_str} -> {num\_float}")

*# Convert integer to float*

int\_to\_float = float(num\_int)

print(f"Integer to Float: {num\_int} -> {int\_to\_float}")

*# Convert float to integer*

float\_to\_int = int(num\_float)

print(f"Float to Integer: {num\_float} -> {float\_to\_int}")

*# Convert integer to string*

int\_to\_str = str(num\_int)

print(f"Integer to String: {num\_int} -> {int\_to\_str}")

*# Convert float to string*

float\_to\_str = str(num\_float)

print(f"Float to String: {num\_float} -> {float\_to\_str}")

*# Convert integer to boolean (0 is False, any other number is True)*

int\_to\_bool = bool(num\_int)

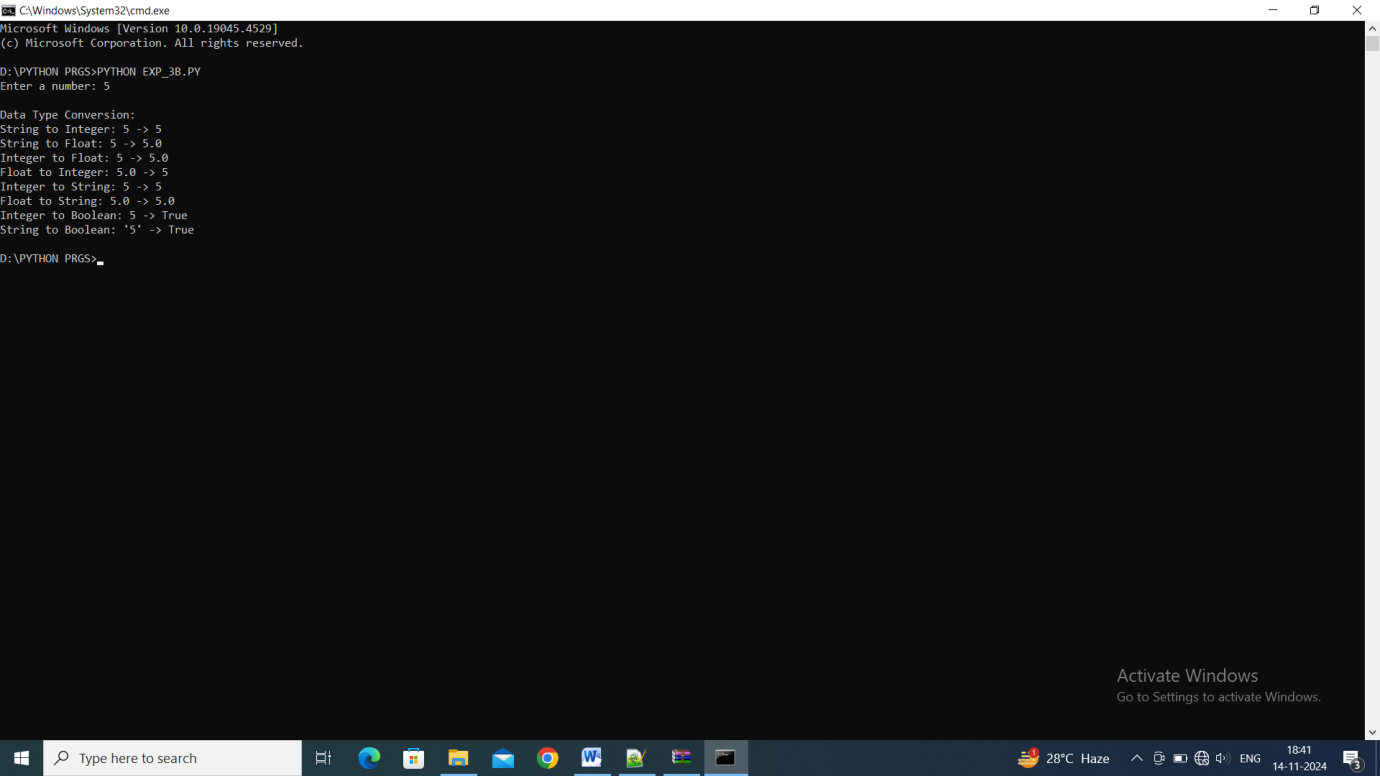
print(f"Integer to Boolean: {num\_int} -> {int\_to\_bool}")

*# Convert string to boolean (empty string is False, any other string is True)*

str\_to\_bool = bool(num\_str)

print(f"String to Boolean: '{num\_str}' -> {str\_to\_bool}")

**OUTPUT:**

****

**4.A. Aim :** Write a program to Convert U.S. dollars to Indian rupees.

**Source Code :**

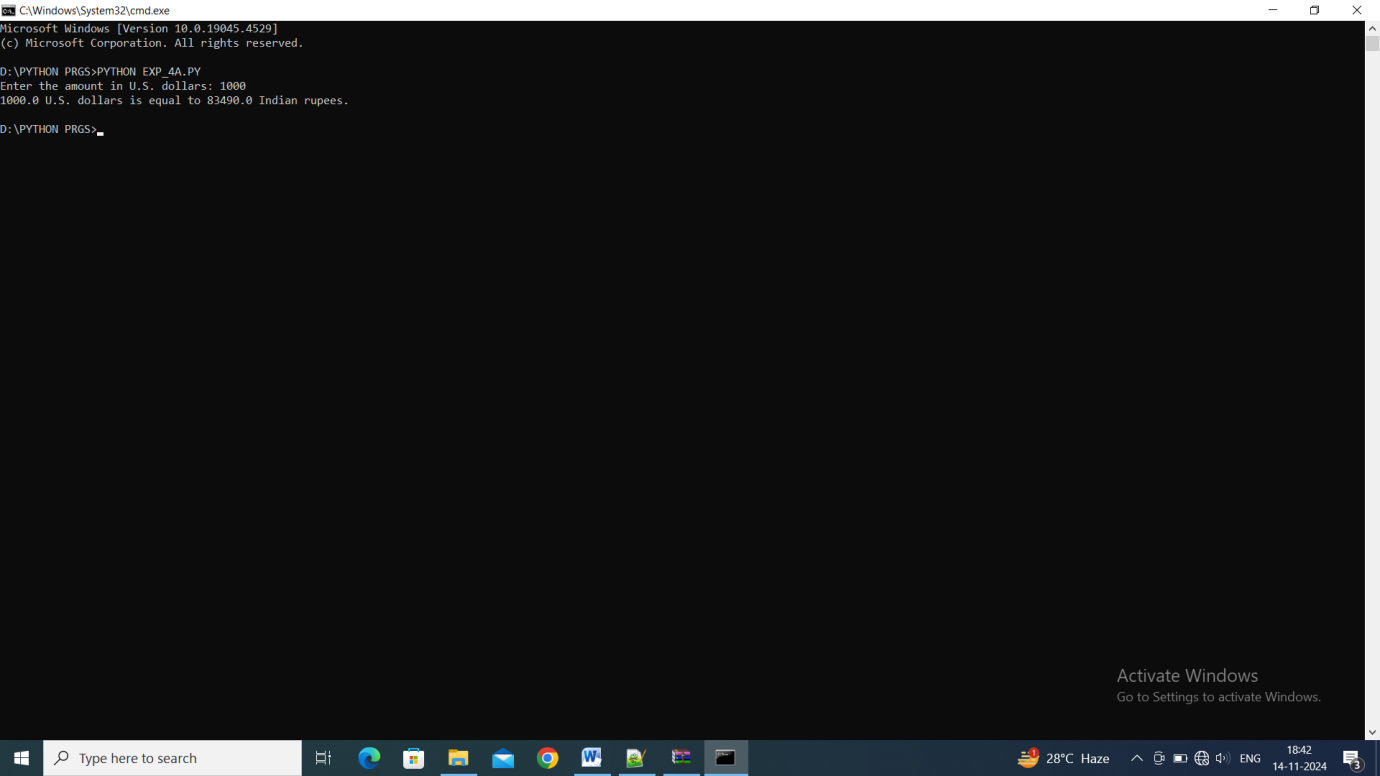
conversion\_rate = 83.49

usd = float(input("Enter the amount in U.S. dollars: "))

inr = usd \* conversion\_rate

print(f"{usd} U.S. dollars is equal to {inr} Indian rupees.")

**OUTPUT:**

****

**4.B. Aim :** Write a program to Convert bits to Megabytes, Gigabytes and Terabytes.

**Source Code :**

BITS\_IN\_BYTE = 8

BYTES\_IN\_KB = 1024

BYTES\_IN\_MB = BYTES\_IN\_KB \* 1024

BYTES\_IN\_GB = BYTES\_IN\_MB \* 1024

BYTES\_IN\_TB = BYTES\_IN\_GB \* 1024

bits = float(input("Enter the number of bits: "))

bytes = bits / BITS\_IN\_BYTE

megabytes = bytes / BYTES\_IN\_MB

gigabytes = bytes / BYTES\_IN\_GB

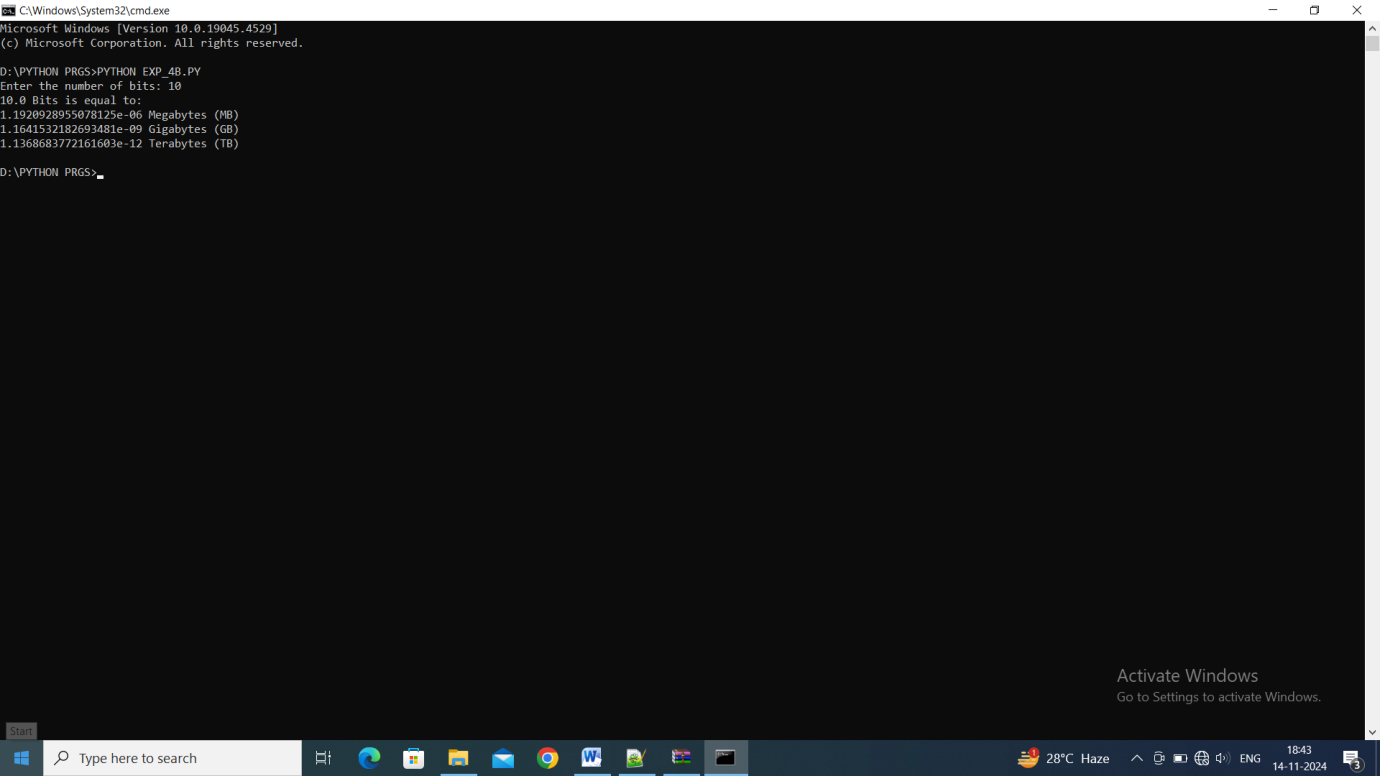
terabytes = bytes / BYTES\_IN\_TB

print(f"{bits} Bits is equal to:")

print(f"{megabytes} Megabytes (MB)")

print(f"{gigabytes} Gigabytes (GB)")

print(f"{terabytes} Terabytes (TB)")



**5.A. Aim :** Write a program to Calculate the area and perimeter of the square.

**Source Code :**

side = float(input("Enter the side of the square: "))

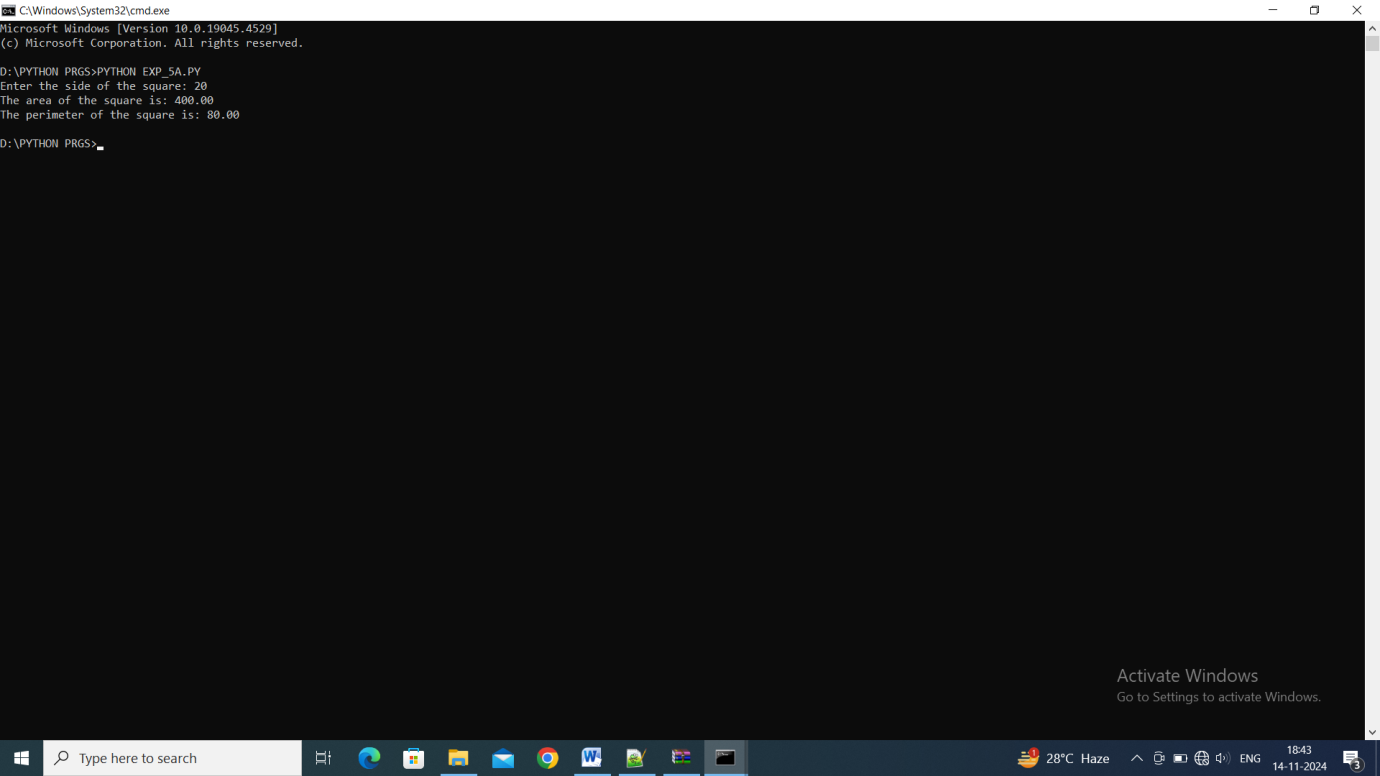
area = side \* side

perimeter = 4 \* side

print(f"The area of the square is: {area:.2f}")

print(f"The perimeter of the square is: {perimeter:.2f}")

**OUTPUT:**

****

**5.B. Aim :** Write a program to Calculate the volume & perimeter of the cone.

**Source Code :**

*import* math

radius = float(input("Enter the radius of the cone: "))

height = float(input("Enter the height of the cone: "))

volume = (1/3) \* math.pi \* radius\*\*2 \* height

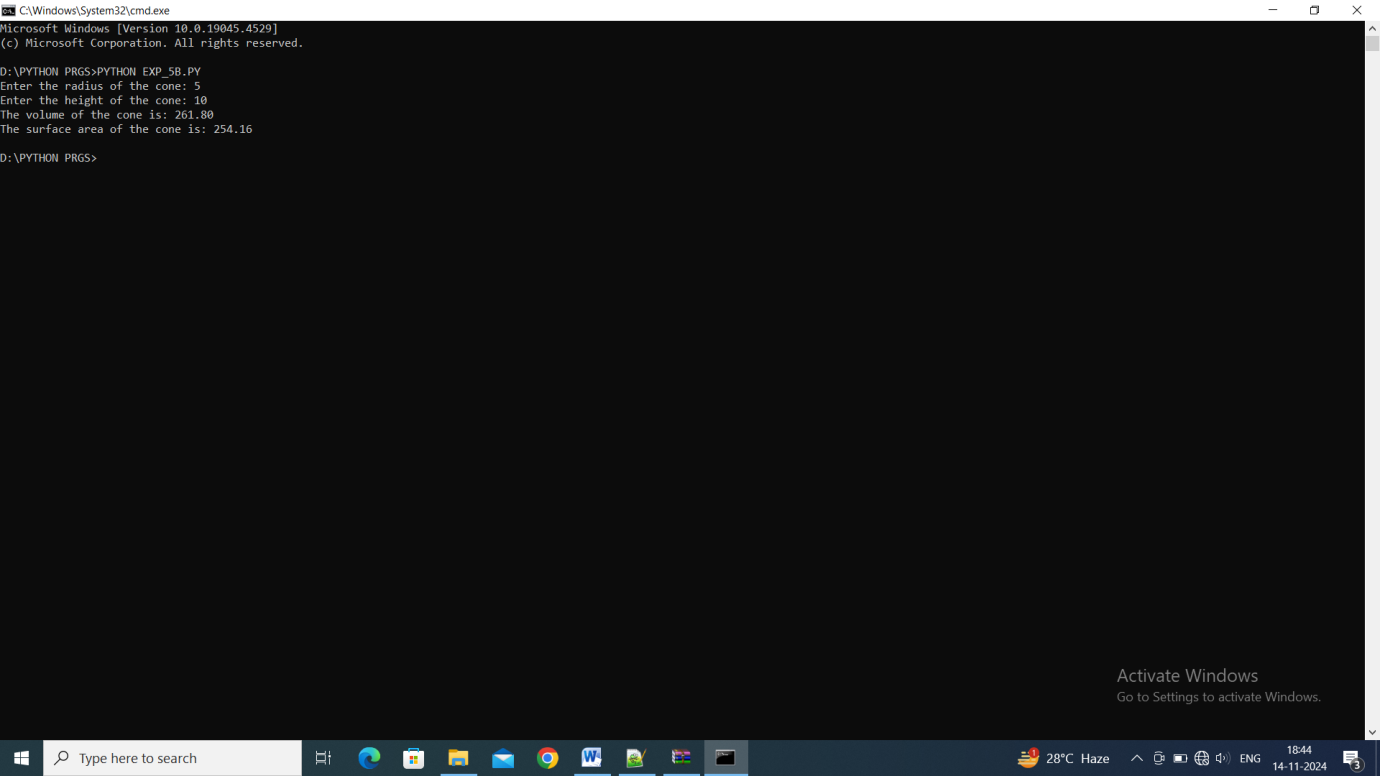
slant\_height = math.sqrt(radius\*\*2 + height\*\*2)

surface\_area = math.pi \* radius \* (radius + slant\_height)

print(f"The volume of the cone is: {volume:.2f}")

print(f"The surface area of the cone is: {surface\_area:.2f}")

**OUTPUT:**

****

**6.A. Aim :** Write a program to Determine whether a given number is odd or even.

**Source Code :**

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

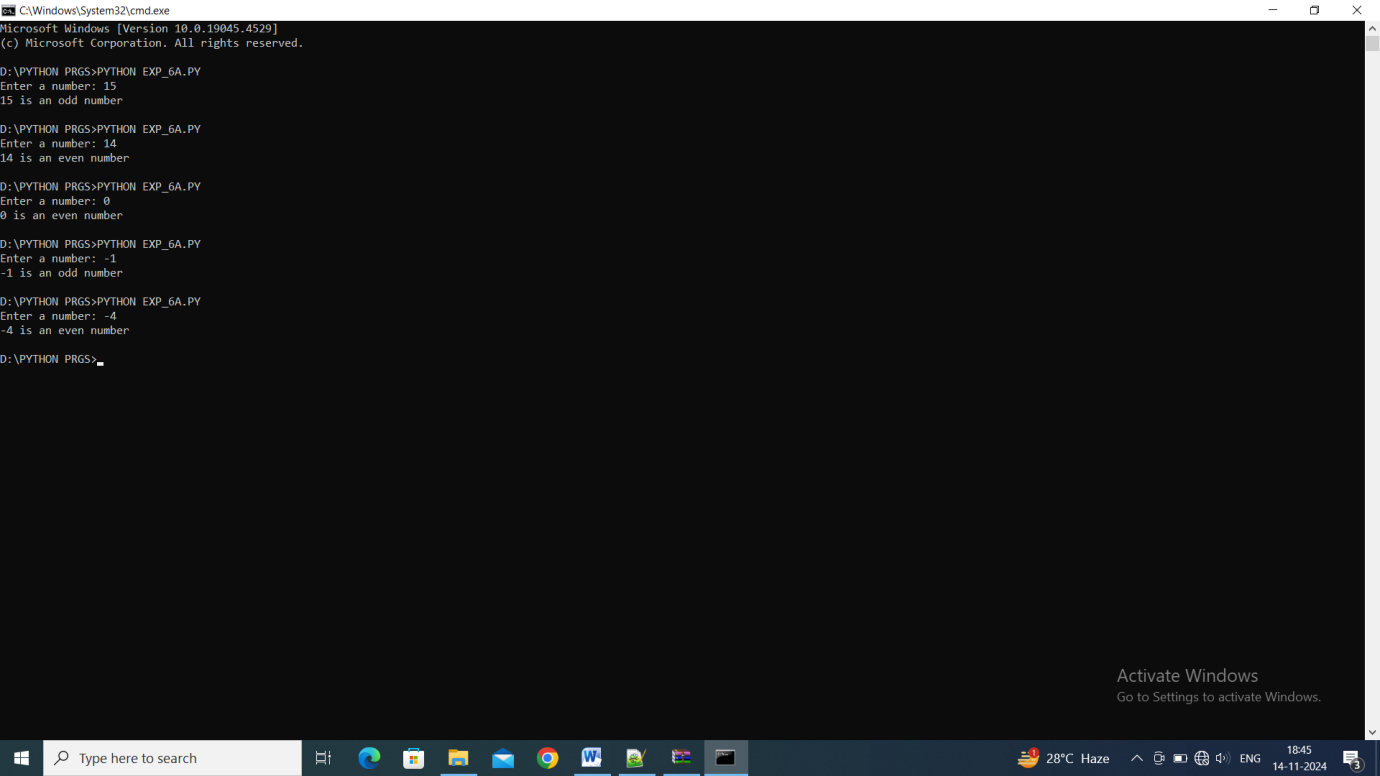
*if* number % 2 == 0:

    print(f"{number} is an even number")

*else*:

    print(f"{number} is an odd number")

**OUTPUT:**

****

**6.B. Aim :** Write a program to Find the greatest of the three numbers using conditional operators.

**Source Code :**

num1 = float(input("Enter the first number: "))

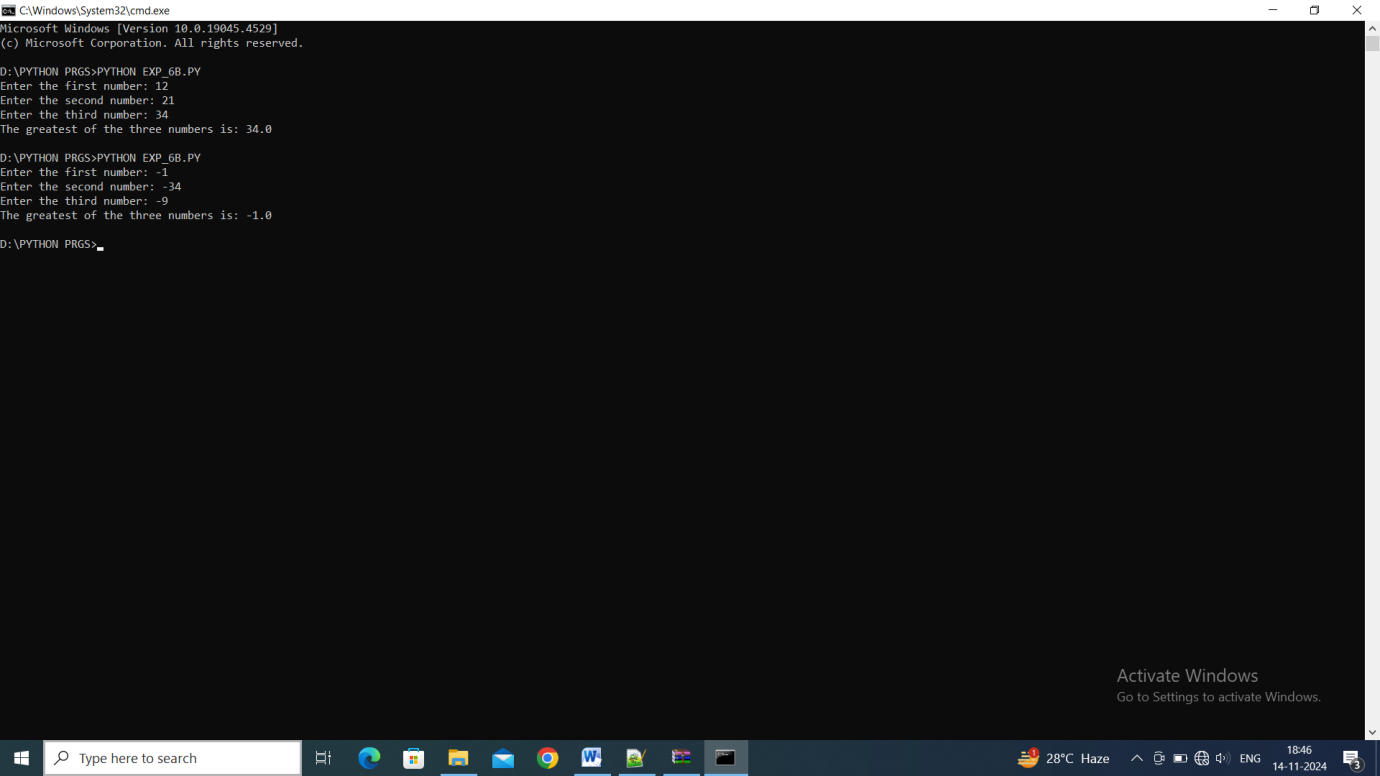
num2 = float(input("Enter the second number: "))

num3 = float(input("Enter the third number: "))

greatest = num1 *if* (num1 > num2 and num1 > num3) *else* (num2 *if* num2 > num3 *else* num3)

print(f"The greatest of the three numbers is: {greatest}")

**OUTPUT:**

****

**7.A. Aim :** Write a program to Find factorial of a given number.

**Source Code :**

number = int(input("Enter a number to find its factorial: "))

factorial = 1

*if* number < 0:

    print("Factorial does not exist for negative numbers")

*elif* number == 0:

    print("The factorial of 0 is 1")

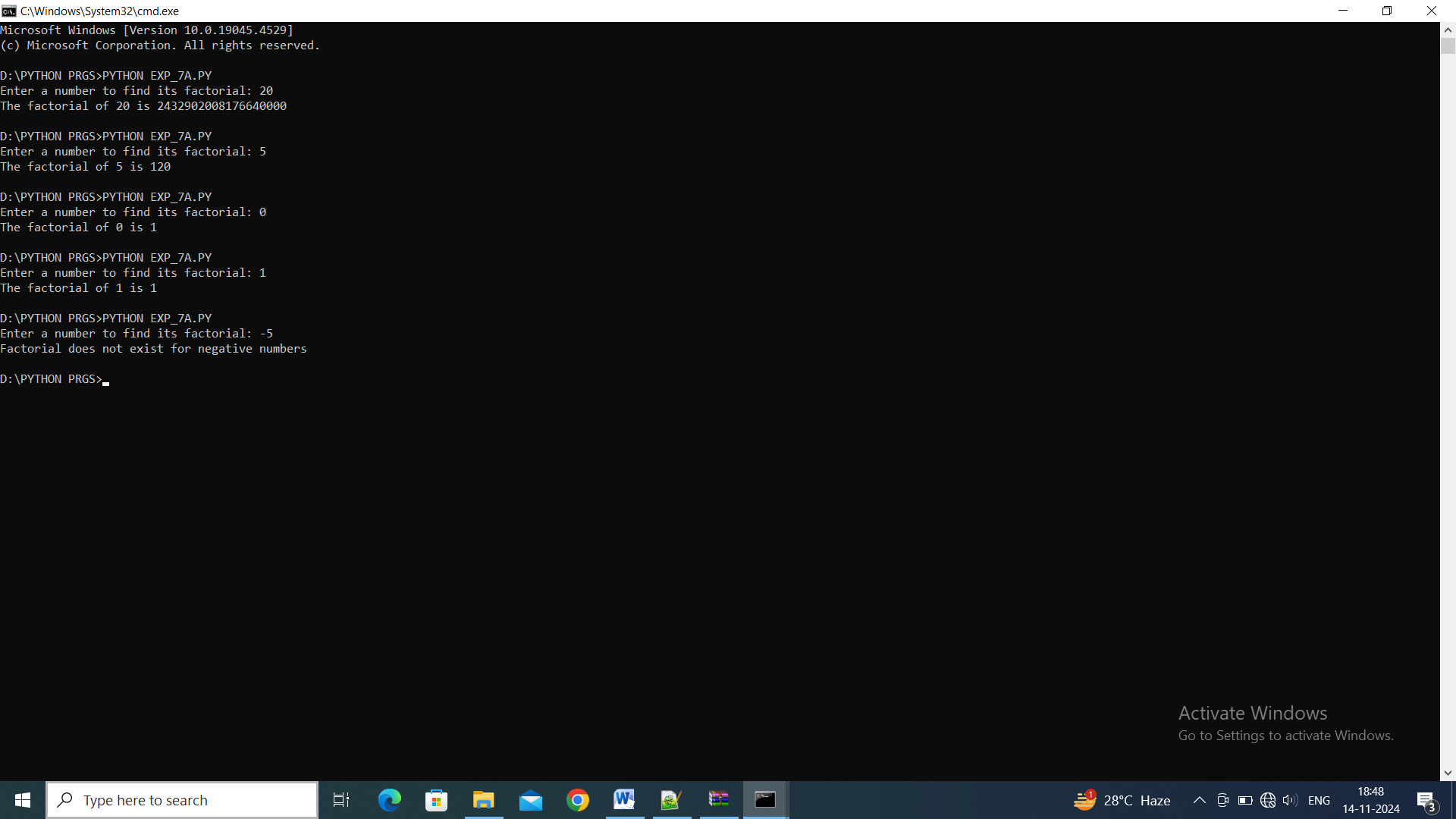
*else*:

*for* i *in* range(1, number + 1):

        factorial \*= i

    print(f"The factorial of {number} is {factorial}")

**OUTPUT:**

****

**7.B. Aim :** Write a program to Generate multiplication table up to 10 for numbers 1 to 5.

**Source Code :**

for num in range(1, 6):

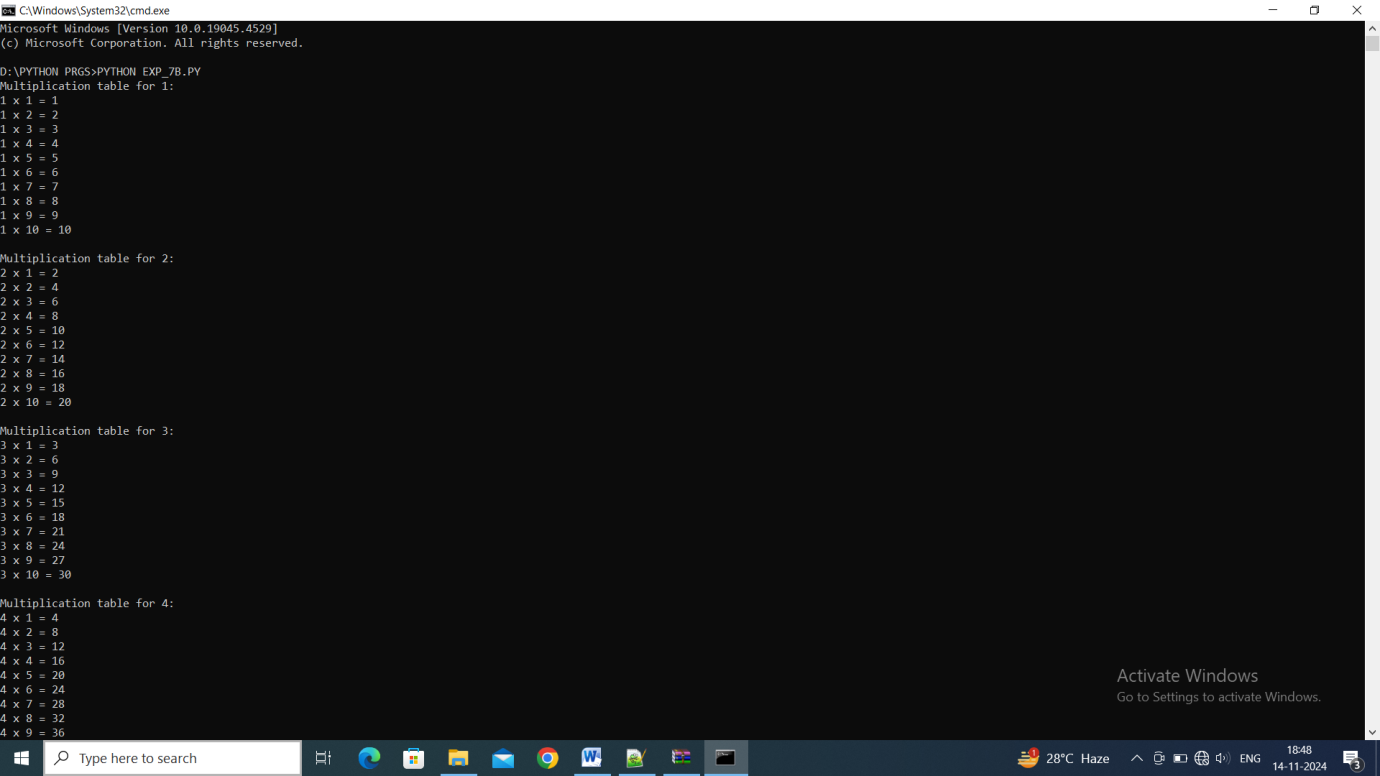
    print(f"Multiplication table for {num}:")

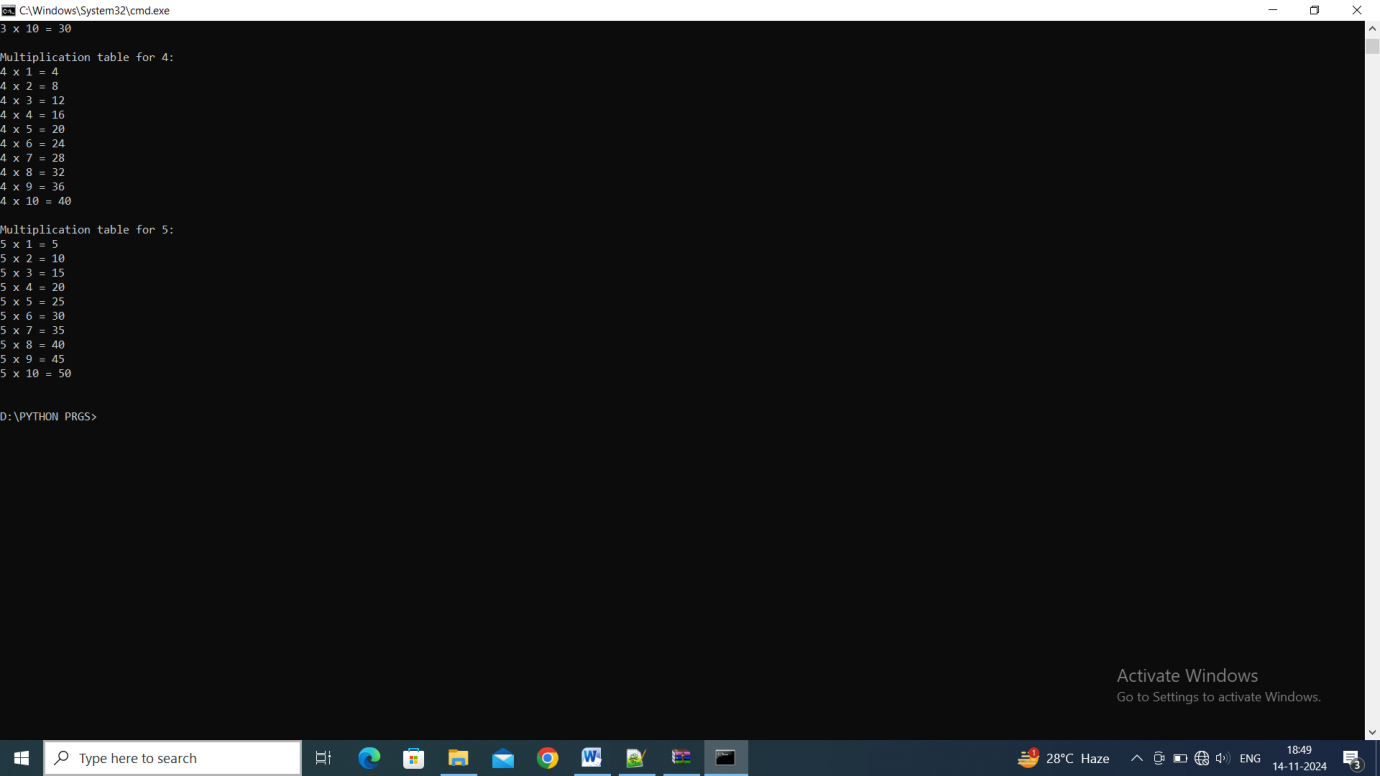
    for i in range(1, 11):

        print(f"{num} x {i} = {num \* i}")

    print()

**OUTPUT:**

****

****

**8.A. Aim :** Write a program Find factorial of a given number.

**Source Code :**

def factorial(*n*):

    result = 1

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

        result \*= i

*return* result

*# Main part of the program*

input\_value = input("Enter a number to find its factorial: ")

*try*:

    num = int(input\_value)

*if* num < 0:

        print("Factorial is not defined for negative numbers.")

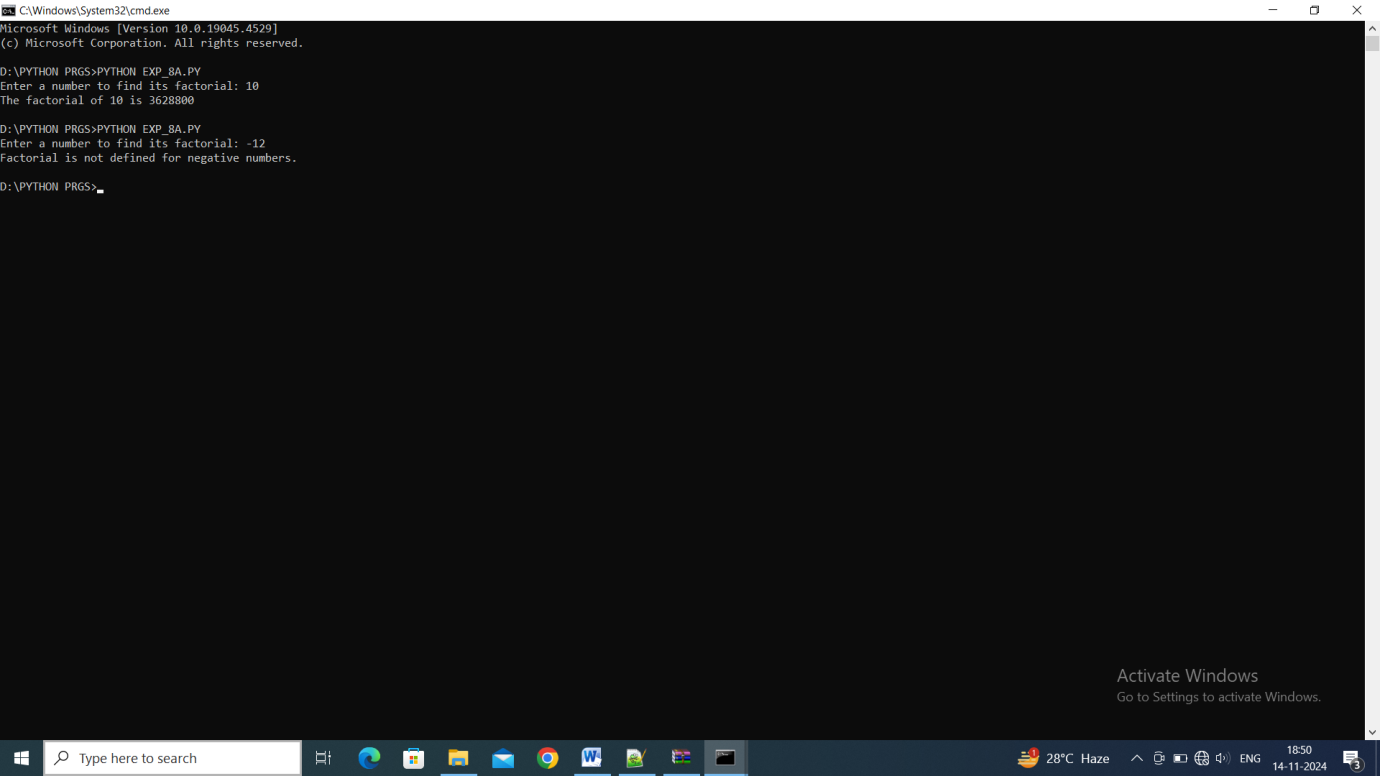
*else*:

        print(f"The factorial of {num} is {factorial(num)}")

*except* ValueError:

    print("Invalid input. Please enter a valid number.")

**OUTPUT:**

****

**8.B. Aim :** Write a program Generate multiplication table up to 10 for numbers 1 to 5 using functions.

**Source Code :**

def multiplication\_table(*start*, *end*):

    i = *start*

*while* i <= *end*:

        print(f"Multiplication table for {i}:")

        j = 1

*while* j <= 10:

            print(f"{i} x {j} = {i \* j}")

            j += 1

        print()

        i += 1

*# Main part of the program*

start\_value = input("Enter a number to start the multiplication table from: ")

end\_value = input("Enter a number to end the multiplication table until: ")

*try*:

    start\_num = int(start\_value)

*if* start\_num <= 0:

        print("Starting number must be greater than 0.")

*else*:

*try*:

            end\_num = int(end\_value)

*if* start\_num <= end\_num:

                multiplication\_table(start\_num, end\_num)

*else*:

                print("Ending number must be greater than or equal to the starting number.")

*except* ValueError:

            print("Invalid input for ending number. Please enter a valid number.")

*except* ValueError:

    print("Invalid input for starting number. Please enter a valid number.")

**9.A. Aim :** Write a program to Find factorial of a given number using recursion.

**Source Code :**

def factorial(*n*):

*if* *n* == 0 or *n* == 1:

*return* 1

*else*:

*return* *n* \* factorial(*n* - 1)

*# Main part of the program*

input\_value = input("Enter a number to find its factorial: ")

*try*:

    num = int(input\_value)

*if* num < 0:

        print("Factorial is not defined for negative numbers.")

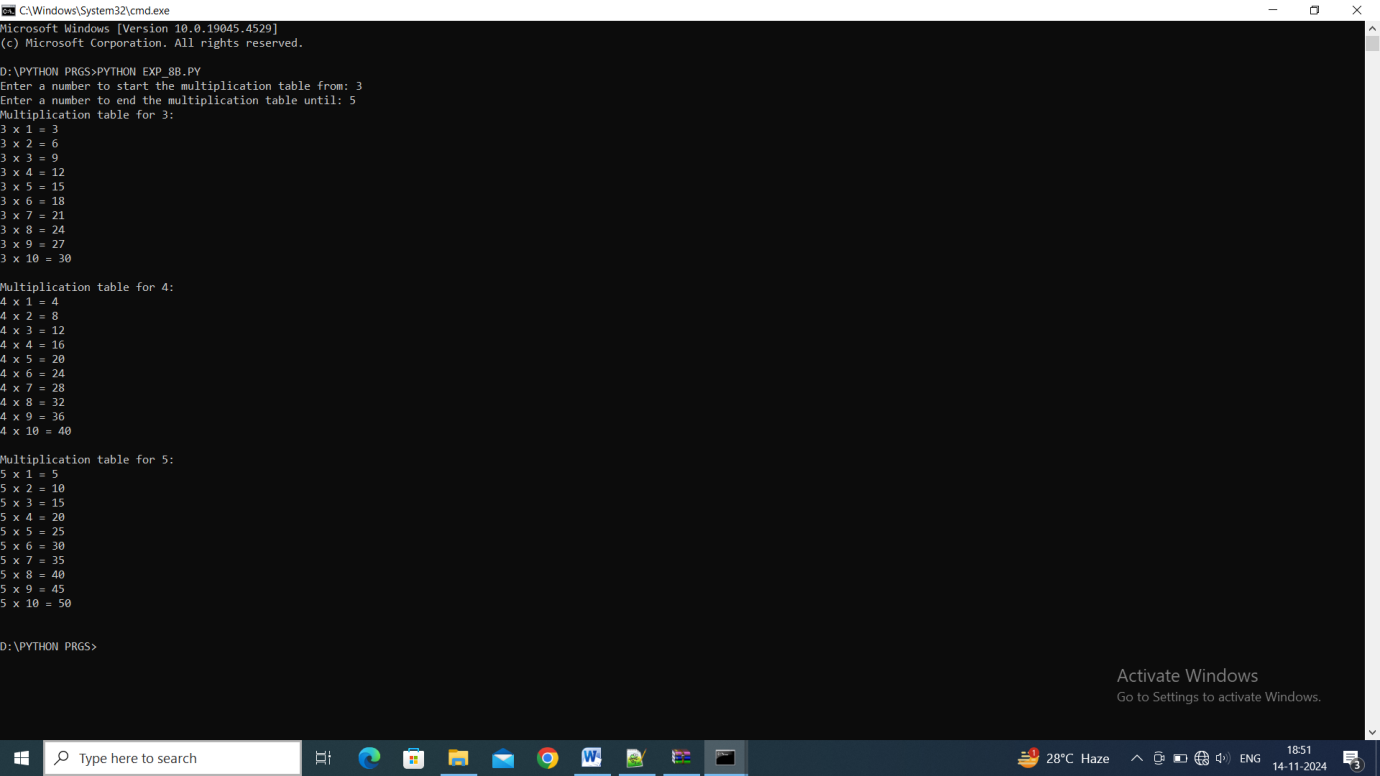
*else*:

        print(f"The factorial of {num} is {factorial(num)}")

*except* ValueError:

    print("Invalid input. Please enter a valid number.")

**OUTPUT:**

****

**9.B. Aim :** Write a program to Generate Fibonacci sequence up to 100 using recursion.

**Source Code :**

def fibonacci(*n*):

*if* *n* <= 1:

*return* *n*

*else*:

*return* fibonacci(*n*-1) + fibonacci(*n*-2)

def print\_fibonacci(*limit*):

    i = 0

*while* True:

        fib\_number = fibonacci(i)

*if* fib\_number > *limit*:

*break*

        print(fib\_number, *end*=" ")

        i += 1

    print()

*# Main part of the program*

*while* True:

    input\_value = input("Enter the maximum value for the Fibonacci sequence (or type 'exit' to quit): ")

*if* input\_value.lower() == 'exit':

        print("Exiting the program. Goodbye!")

*break*

*try*:

        limit = int(input\_value)

*if* limit >= 0:

            print(f"Fibonacci sequence up to {limit}:")

            print\_fibonacci(limit)

            print()

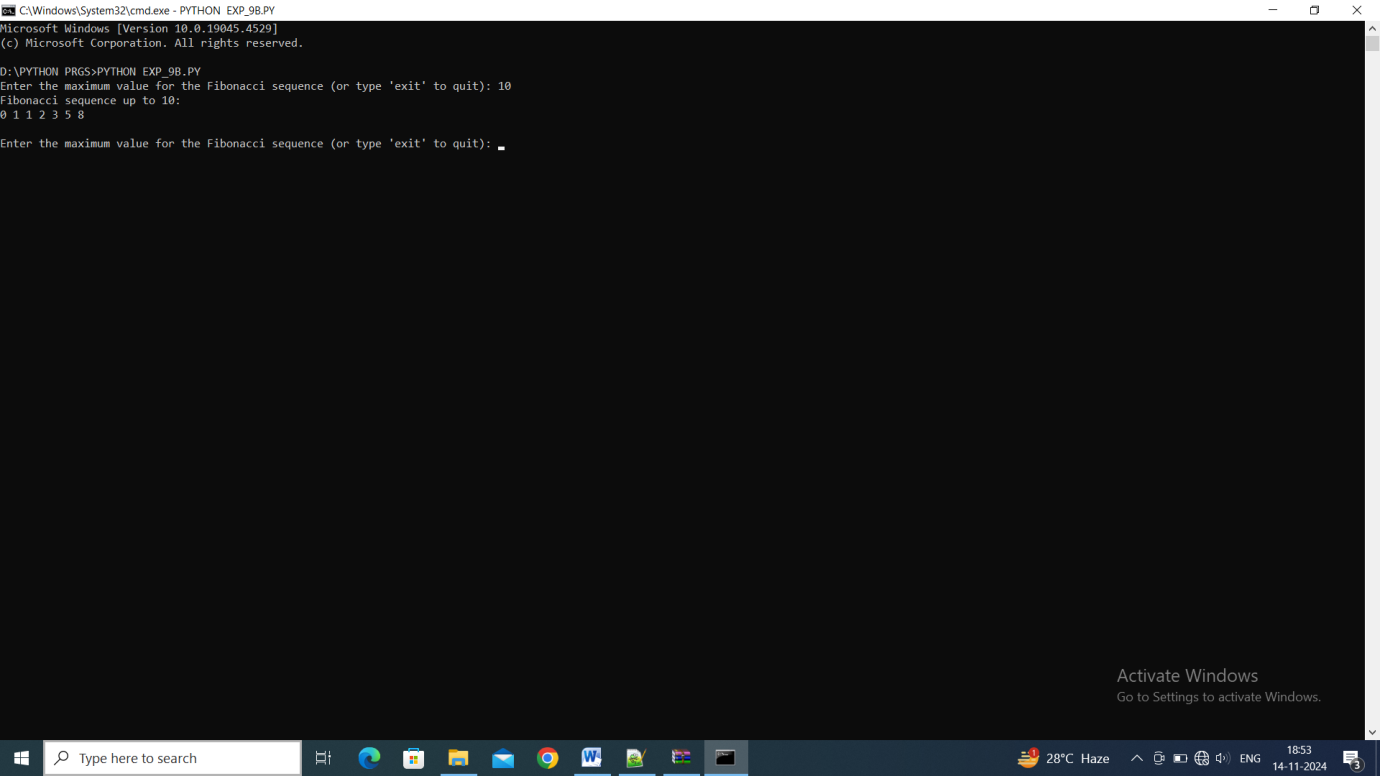
*else*:

            print("Please enter a non-negative integer.\n")

*except* ValueError:

        print("Invalid input. Please enter a valid non-negative integer or type 'exit' to quit.\n")

**OUTPUT:**

****

**10. Aim :** Write a program to:

a. Create a list

b. Add element to the list

c. Delete element from the list

**Source Code :**

def print\_factors(*num*):

    print(f"Factors of {*num*} are:")

*for* i *in* range(1, *num* + 1):

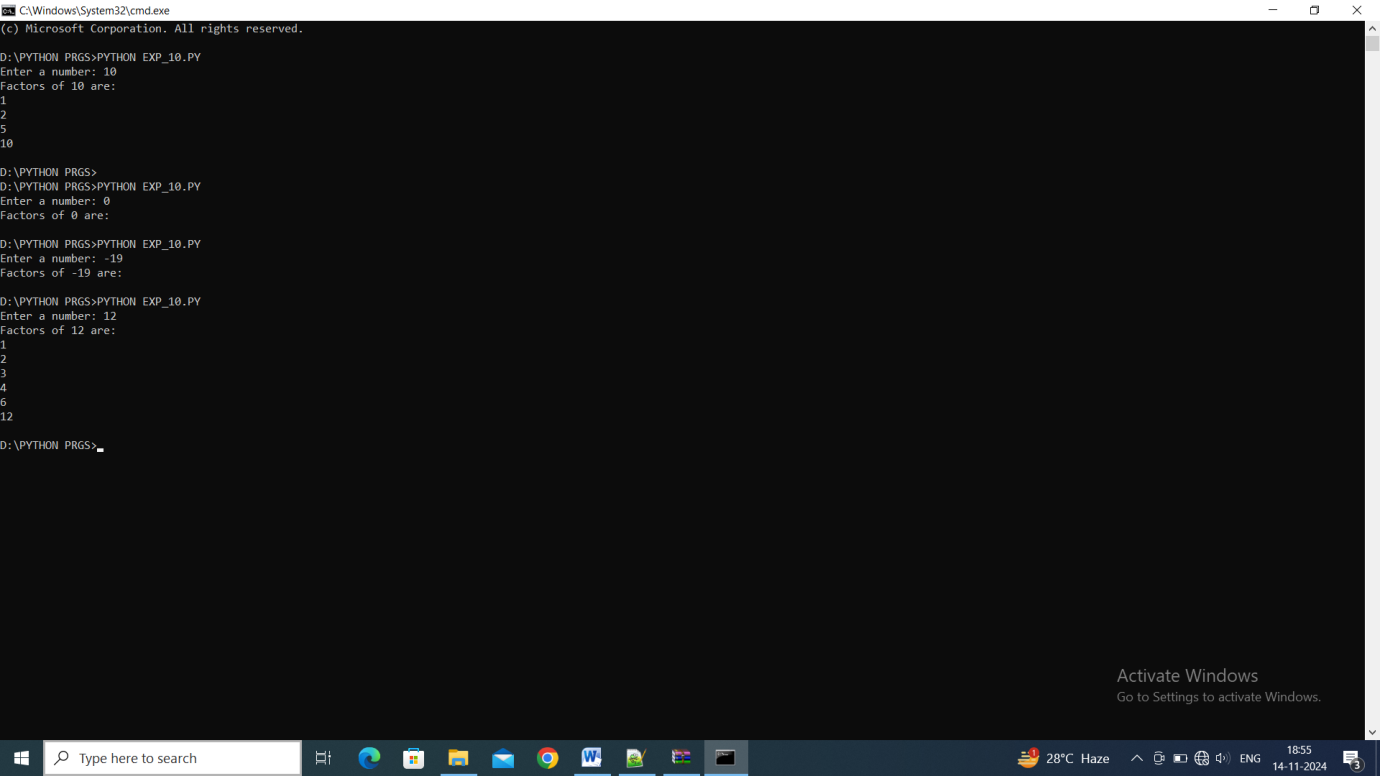
*if* *num* % i == 0:

            print(i)

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

print\_factors(number)

**OUTPUT:**

****

**11. Aim :** . Write a program to:

a. Sort the list

b. Reverse the list

c. Count elements in the list

**Source Code :**

my\_list = []

*while* True:

    print("\nChoose your choice:")

    print("1. Add element")

    print("2. Delete element")

    print("3. Print list")

    print("4. Exit Program")

    choice = int(input("Enter Your choice: "))

*if* choice == 1:

        num\_add = input("Enter the element to add: ")

        my\_list.append(num\_add)

        print(f"{num\_add} added to the list.")

*elif* choice == 2:

*if* my\_list:

            num\_del = input("Enter the element to delete: ")

*if* num\_del in my\_list:

                my\_list.remove(num\_del)

                print(f"{num\_del} removed from the list.")

*else*:

                print(f"{num\_del} not found in the list.")

*else*:

            print("The list is empty. Nothing to delete.")

*elif* choice == 3:

        print("Current list:", my\_list)

*elif* choice == 4:

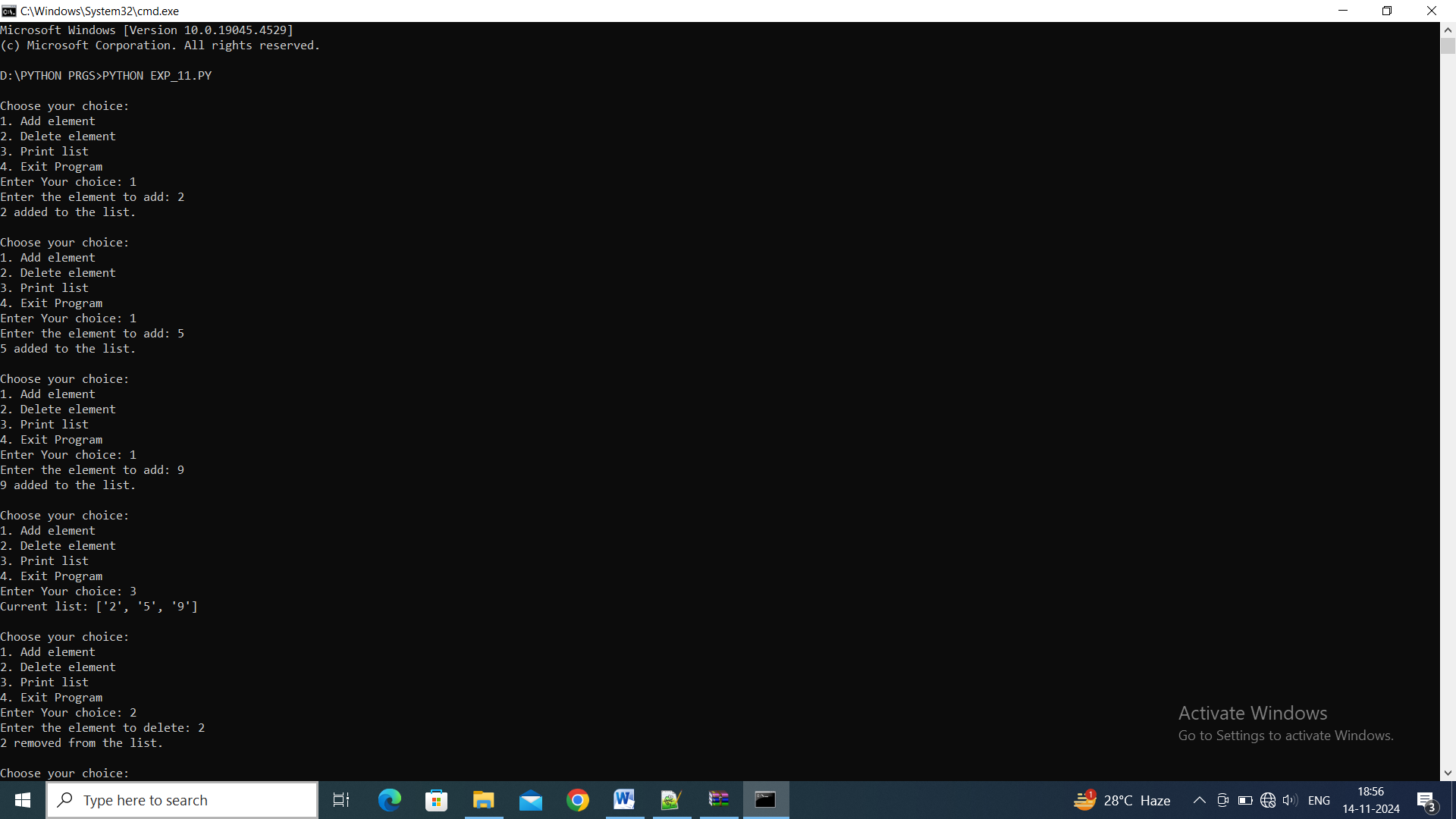
        print("Exiting the program.")

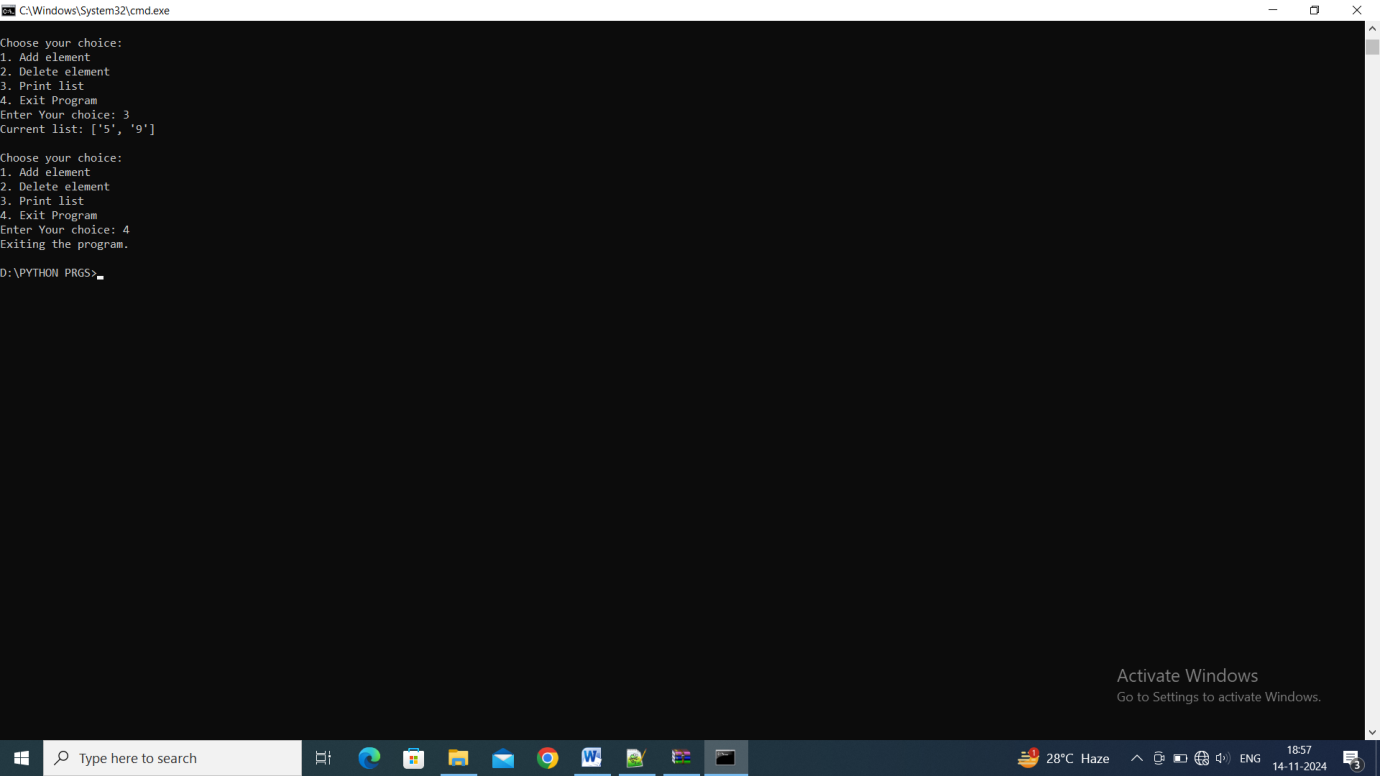
*break*

*else*:

        print("Invalid choice! Please enter a valid option.")

**OUTPUT:**

****

****

**12. Aim :** . Write a program to:

a. Create a dictionary

b. Add element to the dictionary

c. Delete element from the dictionary

**Source Code :**

my\_list = list(map(int, input("Enter elements separated by space: ").split()))

*while* True:

    print("\nChoose your choice:")

    print("1. Sort the list")

    print("2. Reverse the list")

    print("3. Count elements in the list")

    print("4. Exit the program")

    choice = int(input("Your choice: "))

*if* choice == 1:

        my\_list.sort()

        print("Sorted list:", my\_list)

*elif* choice == 2:

        my\_list.reverse()

        print("Reversed list:", my\_list)

*elif* choice == 3:

        print("Total number of elements in the list:", len(my\_list))

*elif* choice == 4:

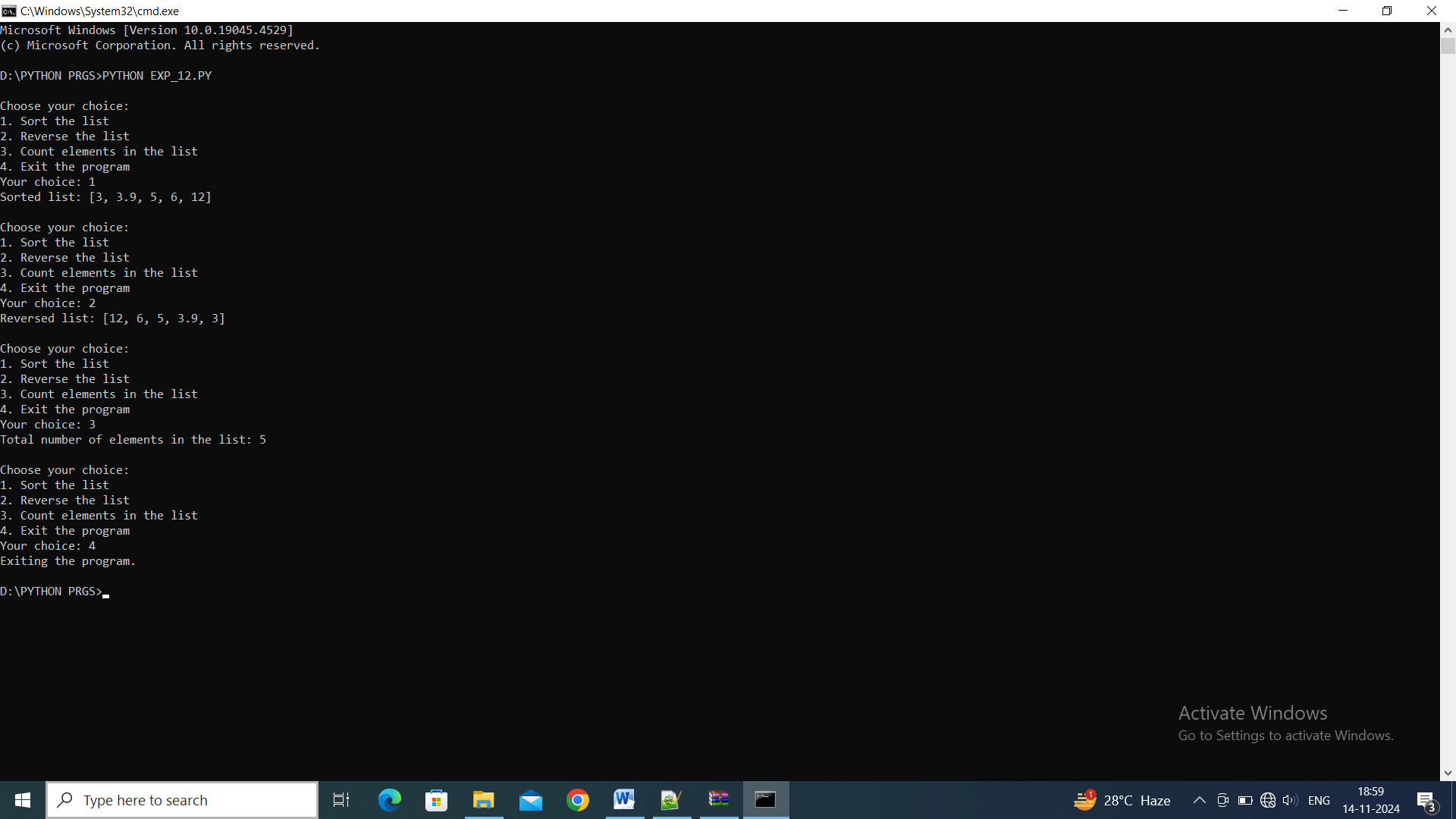
        print("Exiting the program.")

*break*

*else*:

        print("Invalid choice! Please enter a valid option.")

**OUTPUT:**

****

**13. Aim :** . . Write a program to:

a. Calculate average of numbers in a list

b. Calculate mean of numbers in a list

c. Calculate median of numbers in a list

d. Calculate standard deviation of numbers in a list

**Source Code :**

my\_dict = {}

*while* True:

    print("\nChoose your choice:")

    print("1. Add element to the dictionary")

    print("2. Delete element from the dictionary")

    print("3. Print the dictionary")

    print("4. Exit from the program")

    choice = int(input("Enter your choice: "))

*if* choice == 1:

        key = input("Enter the key: ")

        value = input("Enter the value: ")

        my\_dict[key] = value

        print(f"Added ({key}: {value}) to the dictionary.")

*elif* choice == 2:

*if* my\_dict:

            key = input("Enter the key to delete: ")

*if* key in my\_dict:

*del* my\_dict[key]

                print(f"Deleted key '{key}' from the dictionary.")

*else*:

                print(f"Key '{key}' not found in the dictionary.")

*else*:

            print("The dictionary is empty. Nothing to delete.")

*elif* choice == 3:

        print("Current dictionary:", my\_dict)

*elif* choice == 4:

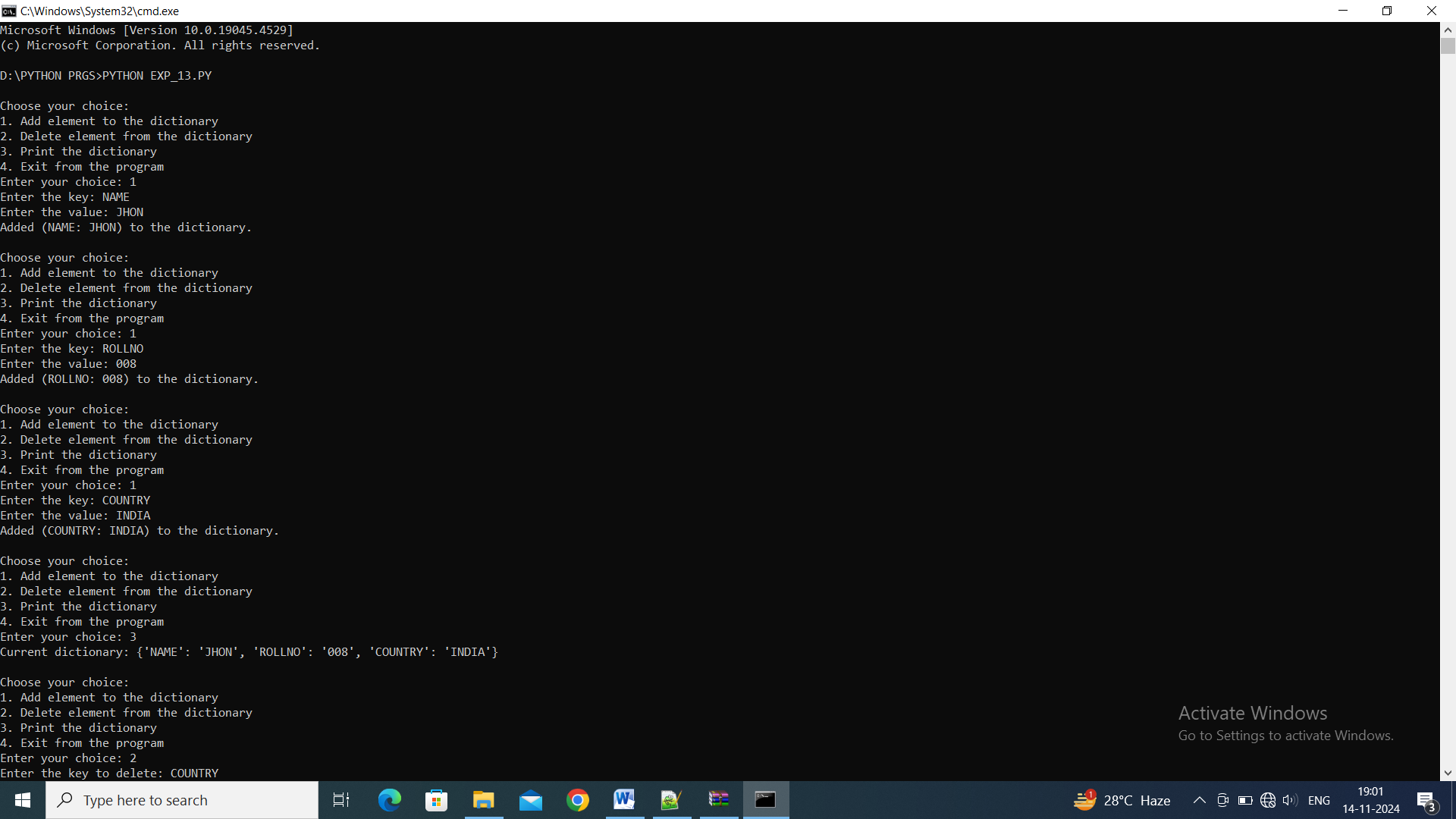
        print("Exiting the program.")

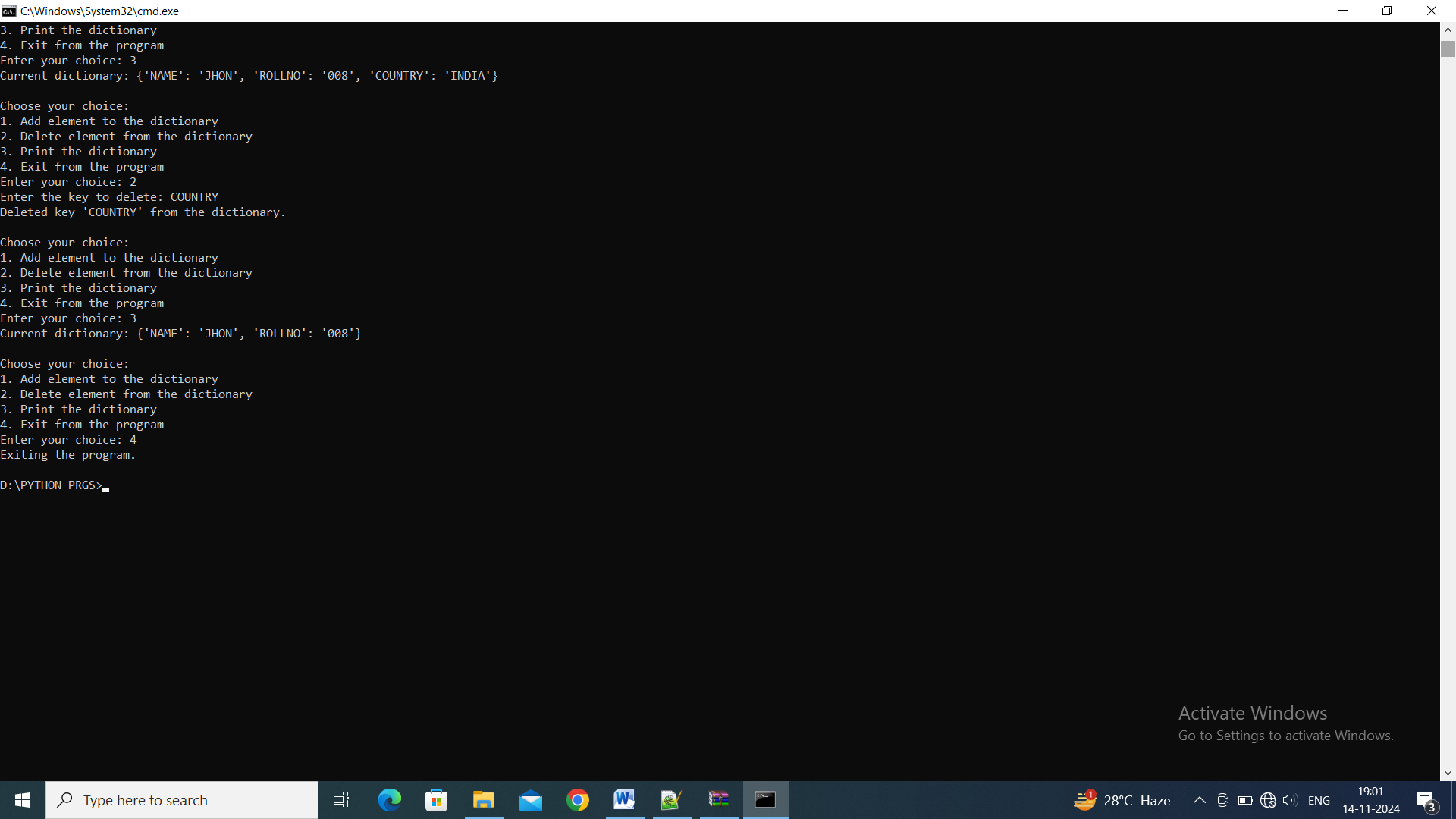
*break*

*else*:

        print("Invalid choice! Please try again.")

**OUTPUT:**

****

****

**14. Aim :** Write a program to: To print factors of a given number

* File Input/output

**Source Code :**

*import* statistics

user\_list = list(map(float, input("Enter elements separated by space: ").split()))

def calculate\_statistics(*numbers*):

*if* len(*numbers*) == 0:

        print("The list is empty. Cannot perform calculations.")

*return*

    average = sum(*numbers*) / len(*numbers*)

    print(f"Average of the numbers: {average}")

    mean = statistics.mean(*numbers*)

    print(f"Mean of the numbers: {mean}")

    median = statistics.median(*numbers*)

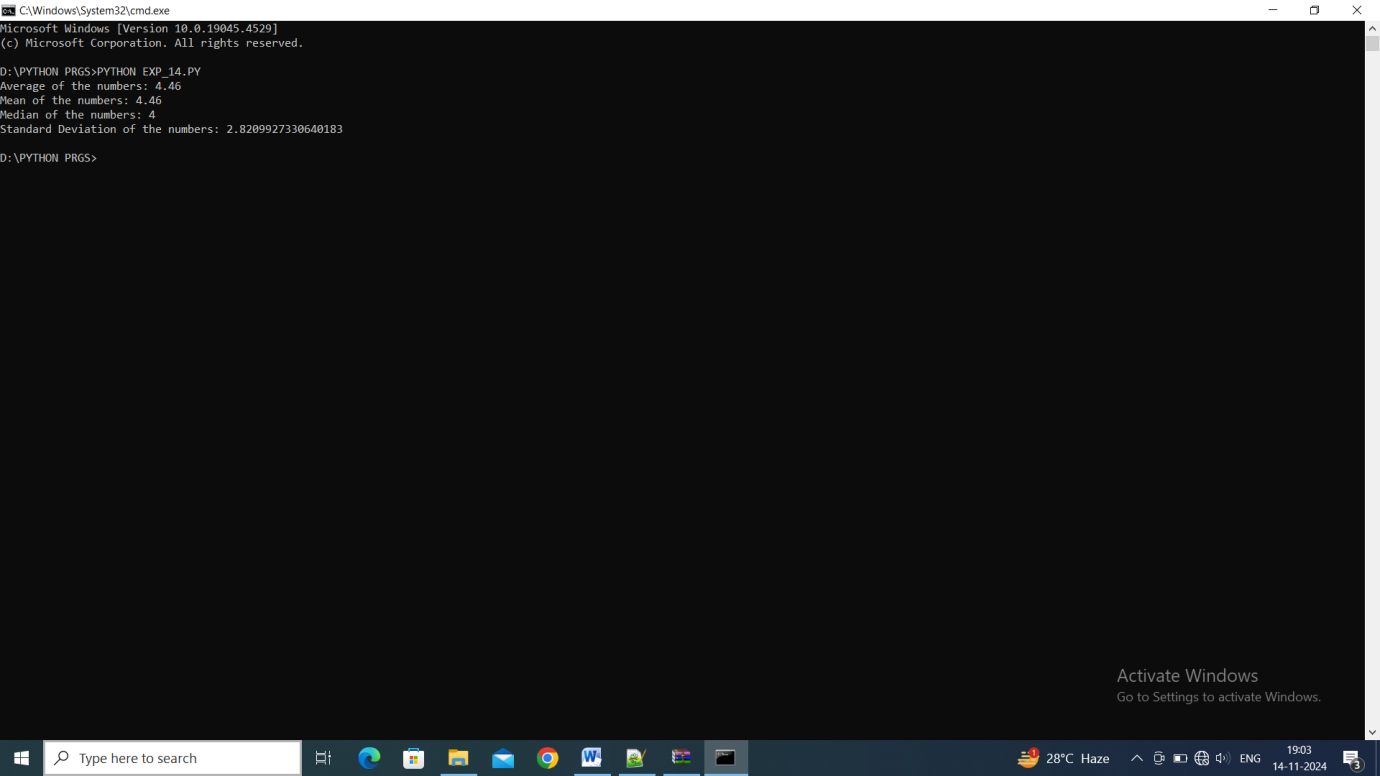
    print(f"Median of the numbers: {median}")

    std\_dev = statistics.stdev(*numbers*)

    print(f"Standard Deviation of the numbers: {std\_dev}")

calculate\_statistics(user\_list)

**OUTPUT:**



**15. Aim :** Write a program to:

i) Create simple file and write “Hello World” in it

ii) Open a file in write mode and append Hello World at the end of a file.

**Source Code :**

*import* os

def create\_file():

*with* open('hello.txt', 'w') *as* file:

        file.write("Hello World")

    print("File(hello.txt) created with Text(Hello World)")

def append\_to\_file():

*with* open('hello.txt', 'a') *as* file:

        file.write("\nHello World")

    print("Text(Hello World) appended into File(hello.txt)")

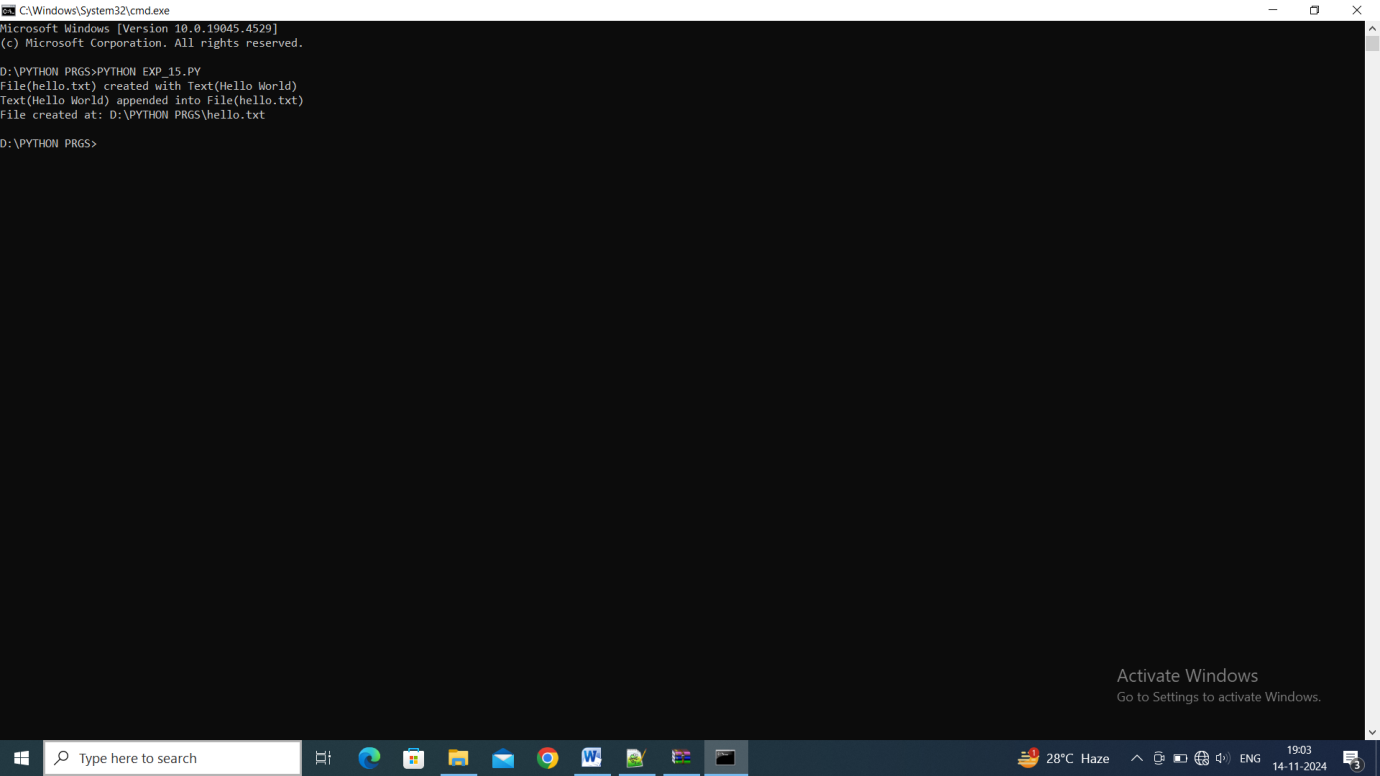
*if* \_\_name\_\_ == "\_\_main\_\_":

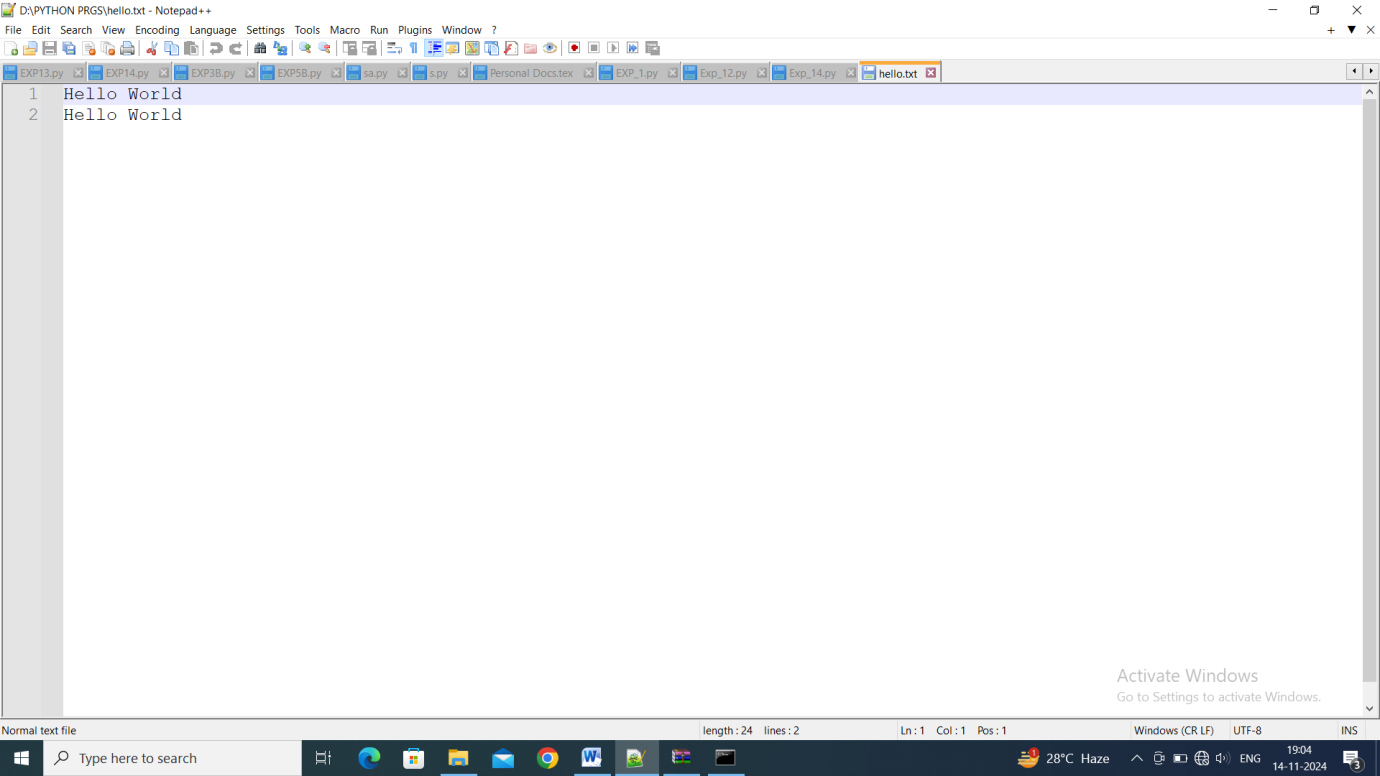
    create\_file()

    append\_to\_file()

    print("File created at:", os.path.abspath('hello.txt'))

**OUTPUT:**

****

****

**16. Aim :** Write a program to :

i) Open a file in read mode and write its contents to another file but replace every occurrence of character ‘h’ .

ii) Open a file in read mode and print the number of occurrences of a character ‘a’.

**Source Code :**

def replace\_character\_in\_file(*source\_file*, *dest\_file*, *char\_to\_replace*, *replacement\_char*):

*with* open(*source\_file*, 'r') *as* src:

        contents = src.read()

        modified\_contents = contents.replace(*char\_to\_replace*, *replacement\_char*)

        modified\_contents = contents.replace('H', *replacement\_char*)

*with* open(*dest\_file*, 'w') *as* dest:

        dest.write(modified\_contents)

    print(f"Contents of '{*source\_file*}' have been written to '{*dest\_file*}' with '{*char\_to\_replace*}' replaced by '{*replacement\_char*}'.")

def count\_character\_in\_file(*source\_file*, *char\_to\_count*):

*with* open(*source\_file*, 'r') *as* src:

        contents = src.read()

        count = contents.count(*char\_to\_count*)

    print(f"The character '{*char\_to\_count*}' occurs {count} times in the file '{*source\_file*}'.")

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

    source\_file = 'source.txt'  *# Replace with your source file*

    dest\_file = 'destination.txt'  *# Replace with your destination file*

    char\_to\_replace = 'h'

    replacement\_char = 'x'

    replace\_character\_in\_file(source\_file, dest\_file, char\_to\_replace, replacement\_char)

    count\_character\_in\_file(source\_file, 'a')

**17. Aim :** Write a Program to: Add two complex number using classes and objects.

**Source Code :**

class ComplexNumber:

    def add\_complex(*self*):

        a = complex(input("Enter first complex number: "))

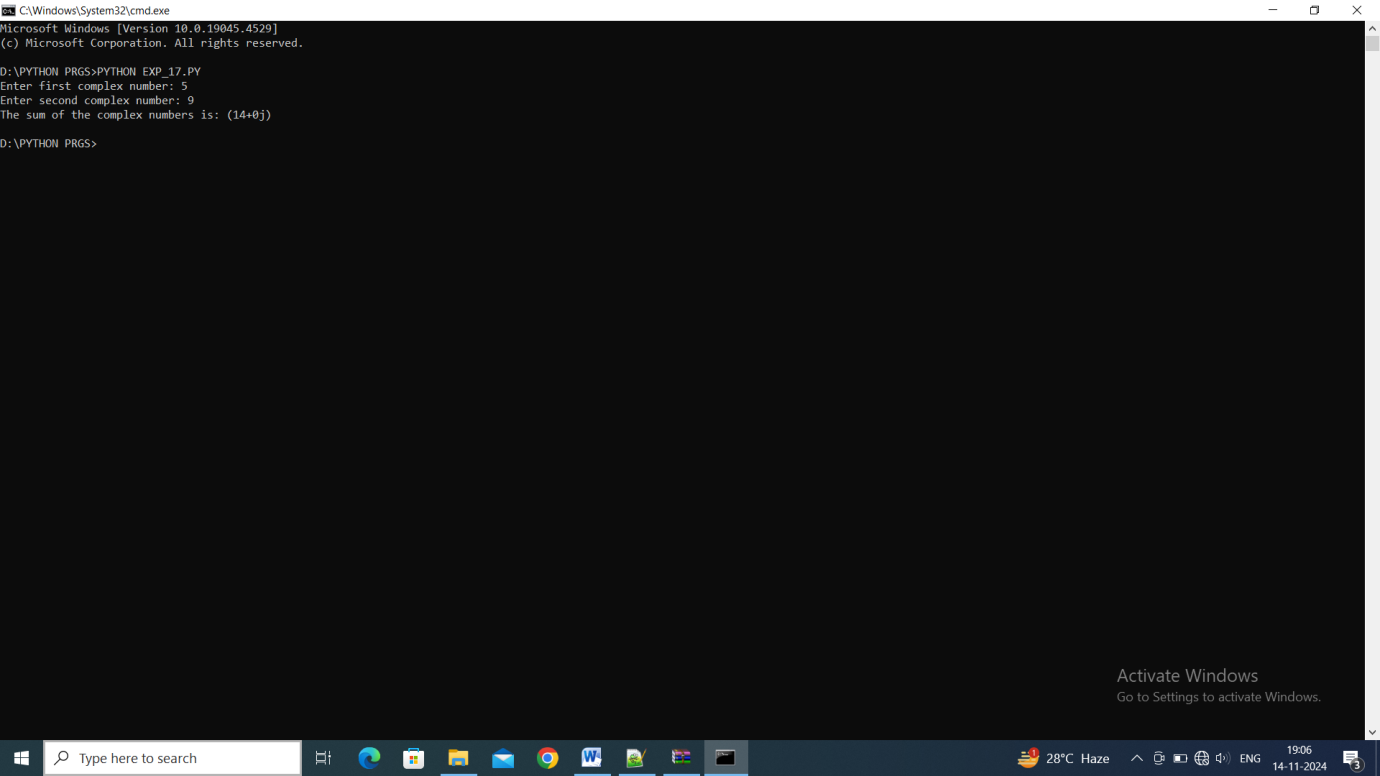
        b = complex(input("Enter second complex number: "))

*return* a + b

obj = ComplexNumber()

result = obj.add\_complex()

print("The sum of the complex numbers is:", result)



**18. Aim :** . Write a Program to: Subtract two complex number using classes and objects.

**Source Code :**

class ComplexNumber:

    def sub\_complex(*self*):

        a = complex(input("Enter first complex number: "))

        b = complex(input("Enter second complex number: "))

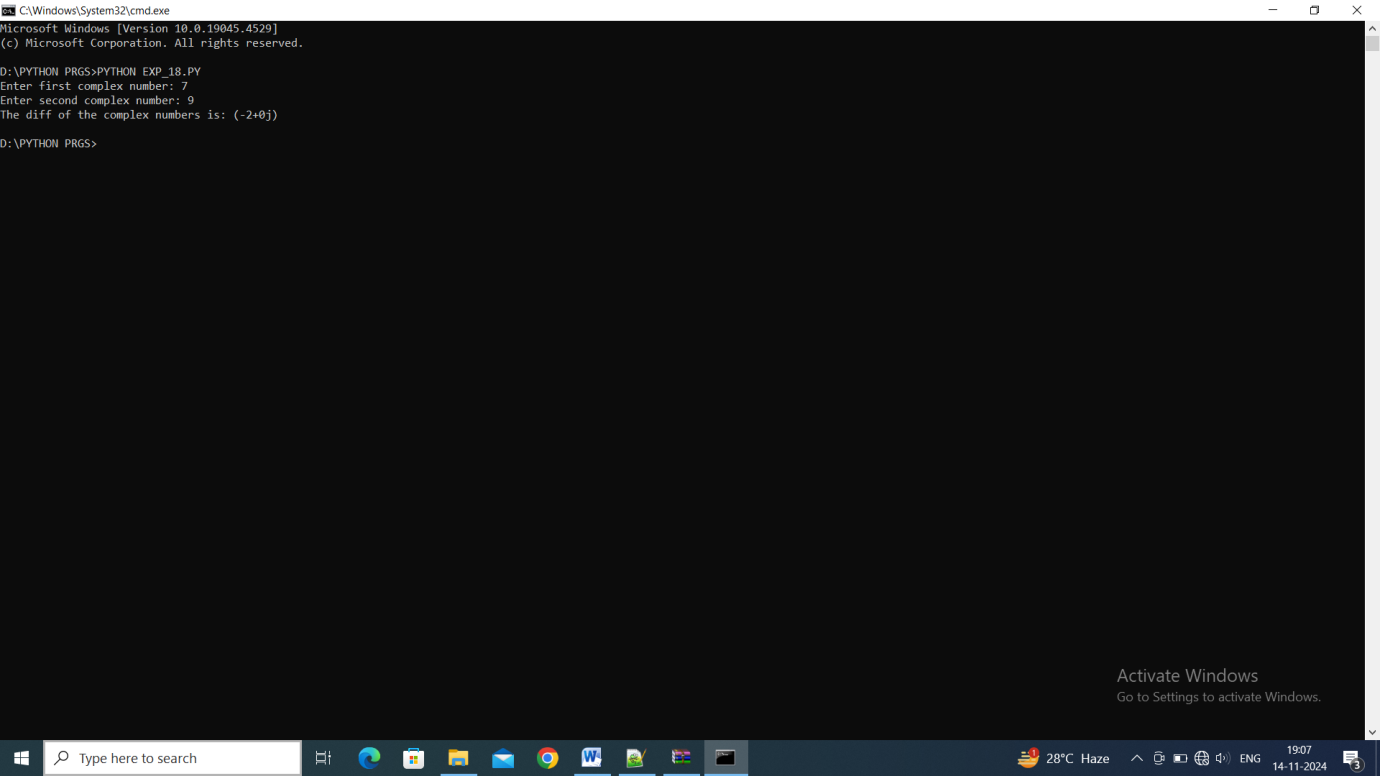
*return* a - b

obj = ComplexNumber()

result = obj.sub\_complex()

print("The diff of the complex numbers is:", result)

**OUTPUT:**



**19. Aim :** Write a Program to: Create a package and accessing a package.

**Source Code :**

*'''*

*Step 1: Create a Package*

*A package in Python is simply a directory that contains a special file \_\_init\_\_.py and one or more modules (Python files).*

*'''*

*#1. Create a directory (folder) named mypackage. Inside this folder, create the following files:*

    mypackage/

    \_\_init\_\_.py

    module1.py

    module2.py

*#2. Add content to these files:*

\_\_init\_\_.py

*# This file can be left empty, or you can add initialization code if needed*

module1.py

def greet():

    print("Hello from module1!")

module2.py

def farewell():

    print("Goodbye from module2!")

*'''*

*Step 2: Access the Package*

*Now, create a Python script that imports and uses the package.*

*'''*

main.py (this file should be outside the mypackage directory)

*# Importing the package*

*from* mypackage *import* module1, module2

*# Accessing functions from the modules*

module1.greet()       *# This will print: Hello from module1!*

module2.farewell()    *# This will print: Goodbye from module2!*

*'''*

*Directory Structure:*

*project/*

*main.py*

*mypackage/*

*\_\_init\_\_.py*

*module1.py*

*module2.py*

*'''*