1.Create a simple calculator in python

def add(num1, num2):

return num1 + num2

def subtract(num1, num2):

return num1 - num2

def multiply(num1, num2):

return num1 \* num2

def divide(num1, num2):

if num2 != 0:

return num1 / num2

else:

return "Error: Cannot divide by zero"

print("Simple Calculator")

print("------------------")

while True:

print("Select operation:")

print("1. Add")

print("2. Subtract")

print("3. Multiply")

print("4. Divide")

print("5. Exit")

choice = input("Enter your choice (1-5): ")

if choice == '5':

print("Exiting the calculator...")

break

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

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

if choice == '1':

print("Result:", add(num1, num2))

elif choice == '2':

print("Result:", subtract(num1, num2))

elif choice == '3':

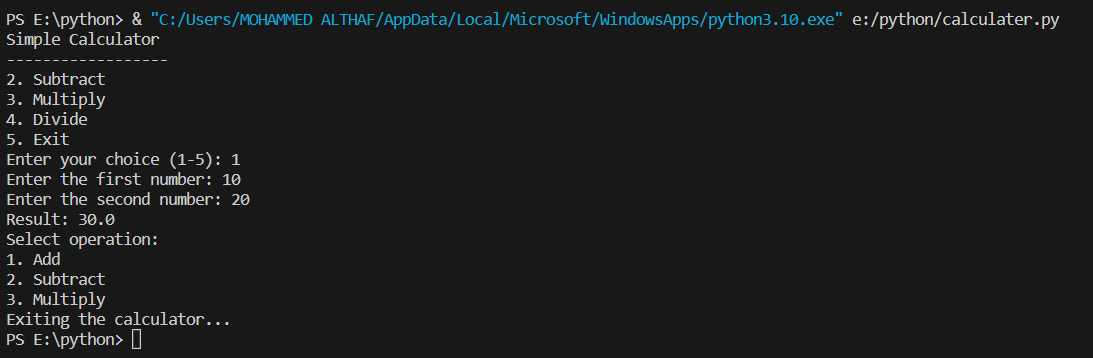
print("Result:", multiply(num1, num2))

elif choice == '4':

print("Result:", divide(num1, num2))

else:

print("Invalid input. Please try again.")



2. An electric power distribution company charges domestic customers as follows:

Consumption unit Rate of charge:

● 0-200 Rs .0.50 per unit.

● 201-400 Rs. 0.65 per unit in excess of 200.

● 401-600 Rs 0.80 per unit in excess of 400.

● 601 and above Rs 1.00 per unit in excess of 600.

● If the bill exceeds Rs. 400, then a surcharge of 15% will be charged, and

the minimum bill should be Rs. 100/-

Create a python program based on the scenario mentioned above.

def calculate\_bill(units):

    total\_bill = 0

    if units <= 200:

        total\_bill = units \* 0.50

    elif units <= 400:

        total\_bill = 200 \* 0.50 + (units - 200) \* 0.65

    elif units <= 600:

        total\_bill = 200 \* 0.50 + 200 \* 0.65 + (units - 400) \* 0.80

    else:

        total\_bill = 200 \* 0.50 + 200 \* 0.65 + 200 \* 0.80 + (units - 600) \* 1.00

    if total\_bill > 400:

        surcharge = total\_bill \* 0.15

        total\_bill += surcharge

    if total\_bill < 100:

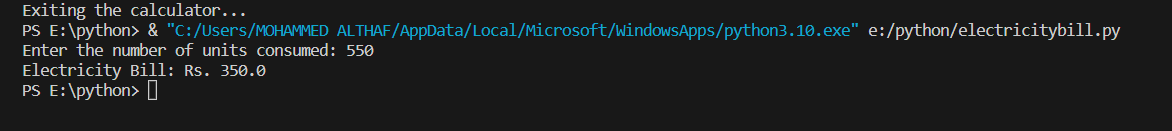
        total\_bill = 100

    return total\_bill

units\_consumed = int(input("Enter the number of units consumed: "))

bill\_amount = calculate\_bill(units\_consumed)

print("Electricity Bill: Rs.", bill\_amount)



3.Print the Pyramid of numbers using for loops.

rows = int(input("Enter the number of rows: "))

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

for j in range(1, rows - i + 1):

print(" ", end="")

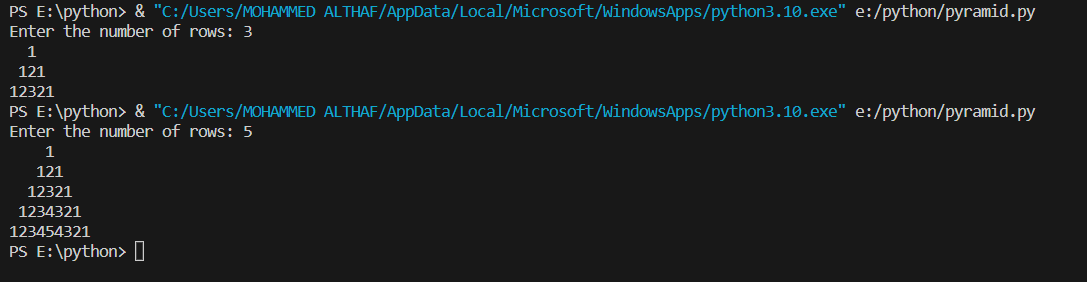
for j in range(1, i + 1):

print(j, end="")

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

print(j, end="")

print()



4.Write a program to find the number of and sum of all integers greater than 100 and less than 200 that are divisible by 7.

count = 0

sum = 0

for num in range(101, 200):

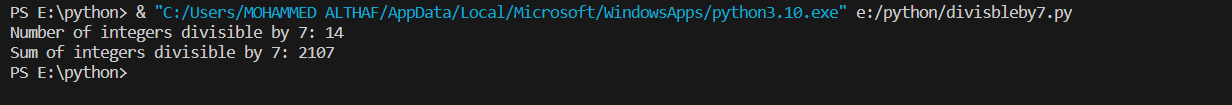
if num % 7 == 0:

count += 1

sum += num

print("Number of integers divisible by 7:", count)

print("Sum of integers divisible by 7:", sum)



5. Write a recursive function to calculate the sum of numbers from 0 to 10

def calculate\_sum(n):

if n == 0:

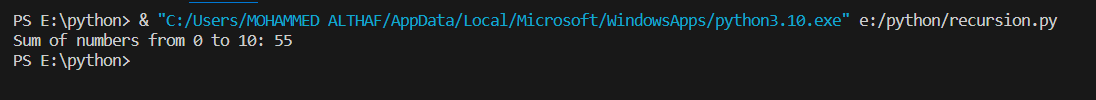
return 0

else:

return n + calculate\_sum(n - 1)

result = calculate\_sum(10)

print("Sum of numbers from 0 to 10:", result)



6.Write a Python program to reverse the digits of a given number and add it to the original, If the sum is not a palindrome repeat this procedure.

def reverse\_number(n):

    return int(str(n)[::-1])

def is\_palindrome(n):

    return str(n) == str(n)[::-1]

def reverse\_and\_add(n):

    while not is\_palindrome(n):

        reverse = reverse\_number(n)

        n += reverse

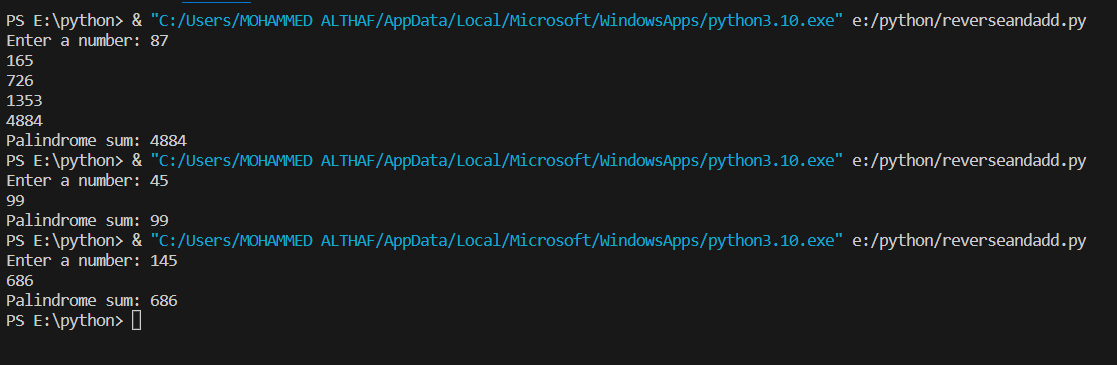
        print(n)

    return n

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

result = reverse\_and\_add(number)

print("Palindrome sum:", result)



7. Write a menu-driven program that performs the following operations on

strings

A) Check String is Substring of Another String

B) Count Occurrences of Character

C) Replace a substring with another substring

D) Convert to Capital Letters

def check\_substring():

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

    substring = input("Enter the substring to check: ")

    if substring in main\_string:

        print(f"'{substring}' is a substring of '{main\_string}'.")

    else:

        print(f"'{substring}' is not a substring of '{main\_string}'.")

def count\_occurrences():

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

    char = input("Enter the character to count: ")

    count = main\_string.count(char)

    print(f"The character '{char}' occurs {count} times in '{main\_string}'.")

def replace\_substring():

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

    old\_substring = input("Enter the substring to replace: ")

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

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

    print("New string:", new\_string)

def convert\_to\_capital():

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

    capitalized\_string = main\_string.upper()

    print("Capitalized string:", capitalized\_string)

while True:

    print("\nMenu:")

    print("A) Check String is Substring of Another String")

    print("B) Count Occurrences of Character")

    print("C) Replace a substring with another substring")

    print("D) Convert to Capital Letters")

    print("E) Exit")

    choice = input("Enter your choice (A/B/C/D/E): ")

    if choice == "A":

        check\_substring()

    elif choice == "B":

        count\_occurrences()

    elif choice == "C":

        replace\_substring()

    elif choice == "D":

        convert\_to\_capital()

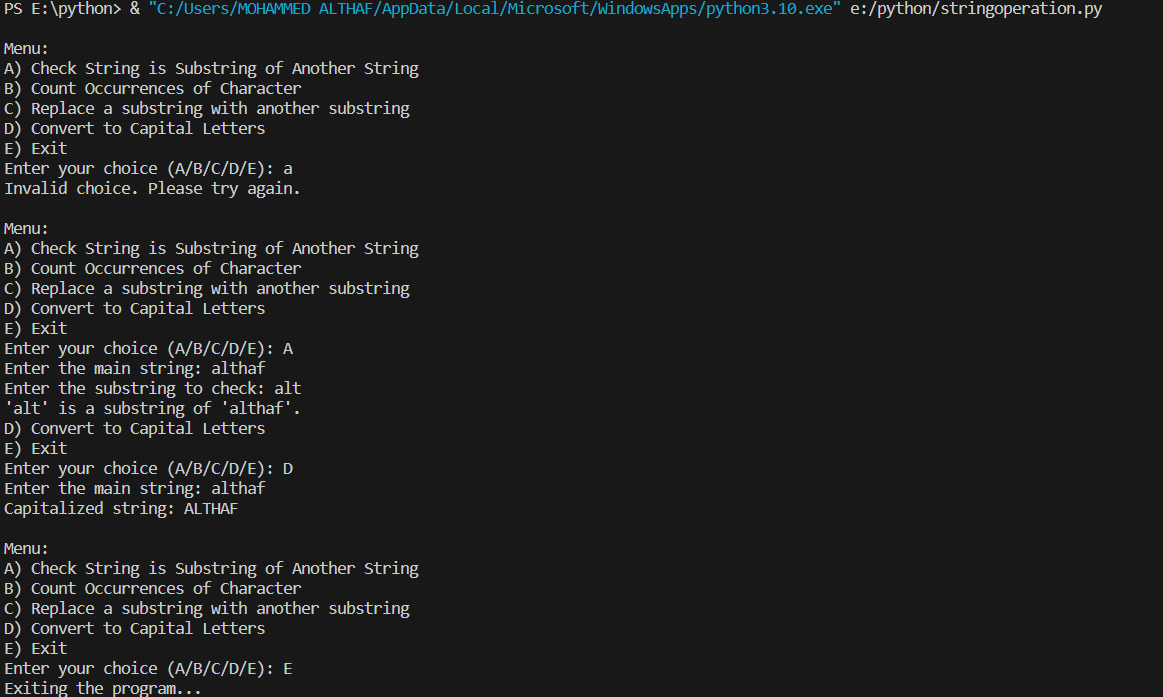
    elif choice == "E":

        print("Exiting the program...")

        break

    else:

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



8. Write a function to find the factorial of a number but also store the factorials

calculated in a dictionary.

factorial\_dict = {}

def factorial(n):

    if n < 0:

        raise ValueError("Factorial is not defined for negative numbers.")

    if n in factorial\_dict:

        return factorial\_dict[n]

    elif n == 0 or n == 1:

        factorial\_dict[n] = 1

        return 1

    else:

        result = 1

        for i in range(2, n + 1):

            result \*= i

            factorial\_dict[i] = result

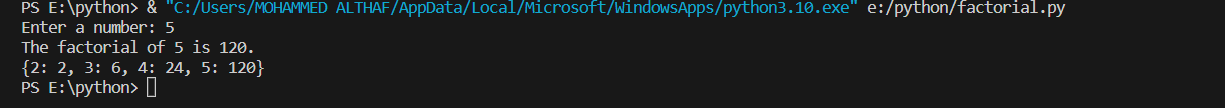
        return result

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

result = factorial(num)

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

print(factorial\_dict)



9.Perform various set operations

A) Set Union

B) Set Intersection

C) Set Difference

set\_A = {1, 2, 3, 4, 5}

set\_B = {4, 5, 6, 7, 8}

union\_set = set\_A.union(set\_B)

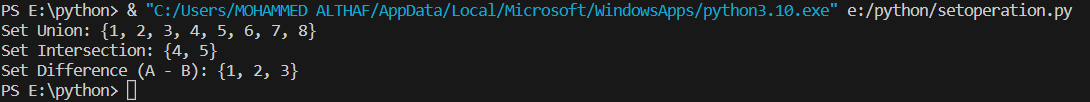
print("Set Union:", union\_set)

intersection\_set = set\_A.intersection(set\_B)

print("Set Intersection:", intersection\_set)

difference\_set = set\_A.difference(set\_B)

print("Set Difference (A - B):", difference\_set)



10. Create a dictionary to store the name, roll\_no, total\_mark of N students. Now

print the details of the student who has got the highest total\_mark.

N = int(input("Enter the number of students: "))

student\_dict = {}

for i in range(N):

    name = input("Enter student name: ")

    roll\_no = input("Enter roll number: ")

    total\_mark = float(input("Enter total marks: "))

    student\_dict[roll\_no] = {'Name': name, 'Total Marks': total\_mark}

highest\_marks = 0

highest\_marks\_student = None

for roll\_no, details in student\_dict.items():

    if details['Total Marks'] > highest\_marks:

        highest\_marks = details['Total Marks']

        highest\_marks\_student = details

print("\nStudent with the highest total marks:")

print("Name:", highest\_marks\_student['Name'])

print("Roll No:", list(student\_dict.keys())[list(student\_dict.values()).index(highest\_marks\_student)])

print("Total Marks:", highest\_marks)

