1.Write a program for selection sorting

Aim:To write a Python program that sorts an array of integers in ascending order using the Selection Sort technique.

```python
def selection_sort(arr):
    n = len(arr)
    for i in range(n):
        min_index = i
        for j in range(i + 1, n):
            if arr[j] < arr[min_index]:
                min_index = j
        arr[i], arr[min_index] = arr[min_index], arr[i]
    return arr
arr = [64, 25, 12, 22, 11]
print("Original Array:", arr)
sorted_arr = selection_sort(arr)
print("Sorted Array:", sorted_arr)
```

```
Original Array: [64, 25, 12, 22, 11]
Sorted Array: [11, 12, 22, 25, 64]

=== Code Execution Successful ===
```

Result:To write a Python program that sorts an array of integers in ascending order using the Selection Sort technique is successfully executed

2.Write a program for duplicate in an instruction.

Aim:To write a Python program that identifies and displays duplicate elements in a list.

```python
def find_duplicates(arr):
    duplicates = []
    for i in arr:
        if arr.count(i) > 1 and i not in duplicates:
            duplicates.append(i)
    return duplicates
arr = [3, 5, 7, 3, 9, 5, 11, 3]
print("Original List:", arr)
dup = find_duplicates(arr)
if dup:
    print("Duplicate Elements:", dup)
else:
    print("No Duplicates Found.")
```

```
Original List: [3, 5, 7, 3, 9, 5, 11, 3]
Duplicate Elements: [3, 5]

=== Code Execution Successful ===
```

Result:To write a Python program that identifies and displays duplicate elements in a list is successfully executed

3.Find the bigger number in a series

Aim:To write a Python program to find the largest (biggest) number in a given list or series of numbers.

```python
def find_biggest(numbers):
    biggest = numbers[0]   # Assume first number is the biggest
    for num in numbers:
        if num > biggest:
            biggest = num
    return biggest
numbers = [45, 22, 89, 16, 72, 54]
print("Series of Numbers:", numbers)
biggest = find_biggest(numbers)
print("The Biggest Number is:", biggest)

```

```
Series of Numbers: [45, 22, 89, 16, 72, 54]
The Biggest Number is: 89

=== Code Execution Successful ===
```

Result:To write a Python program to find the largest (biggest) number in a given list or series of numbers is successfully executed.

4.Write a program for recursion- factorial of a given number

Aim: To write a Python program to find the factorial of a given number using recursion.

```python
def factorial(n):
    if n == 0 or n == 1:
        return 1
    else:
        return n * factorial(n - 1)
num = int(input("Enter a number: "))
result = factorial(num)
print("Factorial of", num, "is:", result)

```

```
Enter a number: 24
Factorial of 24 is: 620448401733239439360000

=== Code Execution Successful ===
```

Result:To write a Python program to find the factorial of a given number using recursion. Is successfully executed

5.Write a program for fibonacci series

Aim:To write a Python program that prints the Fibonacci series up to a given number of terms.

```python
def fibonacci(n):
    a, b = 0, 1
    print("Fibonacci Series:", end=" ")
    for i in range(n):
        print(a, end=" ")
        a, b = b, a + b
n = int(input("Enter the number of terms: "))
fibonacci(n)

```

```
Enter the number of terms: 5
Fibonacci Series: 0 1 1 2 3
=== Code Execution Successful ===
```

Result:To write a Python program that prints the Fibonacci series up to a given number of terms.

6.Write a programme for two order homogenous recursion

Aim: To write a Python program that generates terms of a second-order homogeneous recurrence relation using recursion.

```
def recurrence(n, a, b, T0, T1):
    if n == 0:
        return T0
    elif n == 1:
        return T1
    else:
        return a * recurrence(n - 1, a, b, T0, T1) + b *
            recurrence(n - 2, a, b, T0, T1)
n_terms = int(input("Enter number of terms: "))
a = int(input("Enter coefficient a: "))
b = int(input("Enter coefficient b: "))
T0 = int(input("Enter first term (T0): "))
T1 = int(input("Enter second term (T1): "))
print("\nSequence generated:")
for i in range(n_terms):
```

```
Enter the number of terms: 5
Fibonacci Series: 0 1 1 2 3
=== Code Execution Successful ===
```

Result:To write a Python program that generates terms of a second-order homogeneous recurrence relation using recursion is successfully executed

7.Write a programme for leap year

Aim: To write a Python program to check whether a given year is a leap year or not.

```
year = int(input("Enter a year: "))
if (year % 400 == 0):
    print(year, "is a Leap Year.")
elif (year % 100 == 0):
    print(year, "is Not a Leap Year.")
elif (year % 4 == 0):
    print(year, "is a Leap Year.")
else:
    print(year, "is Not a Leap Year.")
```

```
Enter a year: 2020
2020 is a Leap Year.

=== Code Execution Successful ===
```

Result:To write a Python program to check whether a given year is a leap year or not is successfully executed

8.Find the program for swapping of two numbers

Aim:To write a Python program to swap (exchange) the values of two numbers.

```python
a = int(input("Enter first number: "))
b = int(input("Enter second number: "))
print("\nBefore Swapping:")
print("a =", a, " b =", b)
temp = a
a = b
b = temp
print("\nAfter Swapping:")
print("a =", a, " b =", b)
```

```
Enter first number: 19
Enter second number: 17

Before Swapping:
a = 19  b = 17

After Swapping:
a = 17  b = 19

=== Code Execution Successful ===
```

Result: To write a Python program to swap (exchange) the values of two numbers is successfully executed

9.Write a programme for identifying the palindrome

Aim:To write a Python program to check whether a given string or number is a palindrome or not.

```python
text = input("Enter a string or number: ")
reverse_text = text[::-1]
if text == reverse_text:
    print(text, "is a Palindrome.")
else:
    print(text, "is Not a Palindrome.")

```

```
Enter a string or number: 121
121 is a Palindrome.

=== Code Execution Successful ===
```

Result: the code is successfully executed

10.Write a programme for to find the prime number

Aim:To write a Python program to check whether a given number is prime or not.

```python
num = int(input("Enter a number: "))
if num <= 1:
    print(num, "is not a Prime Number.")
else:
    for i in range(2, int(num**0.5) + 1):
        if num % i == 0:
            print(num, "is not a Prime Number.")
            break
    else:
        print(num, "is a Prime Number.")

```

```
Enter a number: 19
19 is a Prime Number.

=== Code Execution Successful ===
```

Result:To write a Python program to check whether a given number is prime or not is successfully executed