**Few Important Python Programs for**

**Final Semester End Examinations**

1. Write an algorithm to swap two variables with and without using a third variable.
2. Write a python program to check whether a given year is a leap year or not. A year is a leap year if:

It is divisible by 4

If divisible by 100, then it must be divisible by 400.

1. Write a python program to check whether the given input is palindrome or not.
2. Write a python program to display Fibonacci series using recursion.
3. Design a flowchart that recommends travel destinations based on the users budget and preference (adventure or relaxation). If the budget is over Rs 1,50,000, recommend the Himalayas (for Adventure)or Paris (for Relaxation). Otherwise, recommend Goa (for Relaxation) or Waynad (for Adventure).
4. Develop a program that defines several functions to perform different tasks, such as calculating the area of a rectangle, finding the maximum of a list of numbers, and printing a personalized greeting. Analyze how variable scope affects the behavior of these functions by modifying variables inside and outside the functions.
5. Write a Python program to check if it is positive, negative or zero.
6. Simulate a simple game where a user has to guess a number between 1 and 100. Use break and continue statements to control the game flow and provide feedback to the user, ensuring an engaging user experience.
7. Write a python program to print all the even numbers within a given range.
8. Write a program to print the following pattern

\*

\* \*

\* \* \*

\* \* \* \*

\* \* \* \* \*

1. Write a python program to find the factorial of a given number using recursion.
2. Write an algorithm to calculate the sum of the given list of numbers.
3. Write a python program to check if the given number is a perfect number or not.
4. Design a flowchart to print the biggest of the 3 numbers.
5. Write an algorithm to print the individual digits of a given number
6. Design a class Student. Write parameterized constructor \_\_init\_\_ method to initialize the attributes name and rollno. Design the display method to print the values in name and rollno.
7. Create a program that defines a class with both class methods and instance methods. Include methods that perform operations related to the entire class (such as counting instances) and methods that operate on individual instances (such as updating attributes).
8. Write a python program to perform binary search operation.
9. Write an algorithm and a program to print the multiplication table.
10. Write a python program using list comprehension to generate a list of squares of even numbers between 1 and 20.
11. Design a flowchart to reverse the digits of a number.
12. Develop a program that uses set operations to demonstrate union, intersection and difference.
13. Design algorithm and flowchart for bubble sort algorithm.
14. Develop a python program to find the area of a triangle using functions.
15. Create a user defined module with the name my\_module.py. In the my\_module.py, define four functions addition, subtraction, multiplication, division. In the operations.py file, call the functions in the my\_module.py.
16. Write a python program to find the LCM of 2 given numbers using functions.
17. Write a python program to find the GCD of 2 numbers using recursion.
18. String slicing is like cutting a cake into pieces, where different slices represent different parts of the string. Develop a program that takes a sentence as input and prints the first half and the second half separately. Extend the program to handle different slicing techniques, such as slicing by word or by character, and evaluate their effectiveness in different scenarios
19. Think of data structures as different types of storage units in a library. Tuples are like locked bookcases where the contents cannot be changed, lists are like open shelves where books can be added or removed, sets are like bins where only unique items are allowed, and dictionaries are like labelled drawers where each book has a specific label for easy retrieval. Develop a program demonstrating how to store and retrieve data using tuples, lists, sets, and dictionaries. Include examples of adding, removing, and accessing elements, and compare the advantages and limitations of each data structure.
20. In an educational simulation game, different vehicles have different methods of movement (e.g., driving, flying). Create a base Vehicle class and derive specific classes like Car and Airplane that implement their respective movement methods.