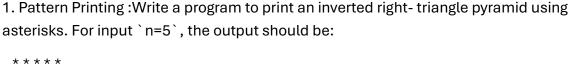
C PROGRAMMING

Control Structures & Loops



* * * * * * * * * * * *

- 2. Alternating Series: Generate the series: `1, -2, 3, -4, 5, -6,...` up to `n` terms using a loop. Ensure the sign alternates between positive and negative.
- 3. Prime Factors: Write a program to print all prime factors of a given number without using `break` or `continue`.

Arrays & Pointers

- 4. Pointer-Based Reversal: Reverse an array of integers using only pointers (no array subscript notation).
- 5. Matrix Diagonal Sum : Calculate the sum of the main diagonal elements of a 3x3 matrix using nested loops.
- 6. Dynamic Array: Use dynamic memory allocation to create an array of `n` integers, populate it with user input, and find the largest element.

Strings & Functions

- 7. Palindrome Check for Strings : Determine if a string is a palindrome using pointers (e.g., `"madam"` is a palindrome).
- 8. Vowel Counter: Write a function that counts the number of vowels (a, e, i, o, u) in a string and returns the count.
- 9. String Concatenation: Implement a function to concatenate two strings without using the `strcat()` library function.

Recursion & Algorithms

- 10. Recursive Fibonacci: Print the Fibonacci sequence up to `n` terms using recursion.
- 11. Recursive Factorial Sum : Compute the sum of the series `1/1! + 1/2! + 1/3! + ... + 1/n!` using recursion.
- 12. Binary Search: Implement a binary search algorithm on a sorted integer array using recursion.

Structures & File Handling

- 13. Student Record System**: Create a `Student` structure with fields `name`, `roll`, and `marks`. Read data for 10 students from a file and print those with marks above 80.
- 14. File Word Counter: Write a program to read a text file and count the number of words, lines, and characters.
- 15. Bitwise Operations: Check if a number is a power of two using bitwise operators (e.g., `8` is `2^3`, so return `true`).