DSA-107 Data Structure and Algorithm Analysis and Design
Recursive Programming

Assignment 1: Writing Recursive Programs (10 Points)

1. Sum of Odd Numbers from 1 to N

Write a recursive function to calculate the sum of all odd numbers from 1 to N.

- **Input Example:** $N = 7$
- **Output:** $1 + 3 + 5 + 7 = 16$

2. Reverse a String

Write a recursive function to reverse a given string.

- **Input Example:** "arun"
- **Output:** "nura"

3. Find the Greatest Common Divisor (GCD)

Write a recursive function to find the GCD of two integers using the Euclidean algorithm.

- **Input Example:** $a = 48, b = 18$
- **Output:** $GCD = 6$

4. Count Vowels in a String

Write a recursive function to count the number of vowels in a given string.

- **Input Example:** "hello"
- **Output:** 2 (vowels: e, o)

5. Exponentiation

Write a recursive function to calculate a^b (a raised to the power of b).

- **Input Example:** $a = 2, b = 4$
- **Output:** 16

6. Check if a String is a Palindrome

Write a recursive function to check if a string is a palindrome (reads the same forward and backward).

- **Input Example:** "radar"
- **Output:** True

[radar, level, madam, refer, civic, deified, racecar, rotor, noon, stats, etc.]

7. Count Occurrences of a Character

Write a recursive function to count the number of times a specific character appears in a string.

- **Input Example:** "mississippi", 's'
- **Output:** 4

8. Generate All Subsets of a String

Write a recursive function to generate all subsets (or power set) of a given string.

- **Input Example:** "abc"
- **Output:** ["", "a", "b", "c", "ab", "ac", "bc", "abc"]

Instructions:

1. Write each program in Python using recursion.
2. Include a base case and recursive case in every program.
3. Test your code with various inputs.