**Converting an Integer to Binary**

Iteration: A while loop was used to repeatedly divide the number by 2 until it becomes 0.

Time complexity: O(log n), each iteration does constant amount of work making it O(log n), where n is the value of the input integer

Recursion: The function calls itself multiple times until base case is reached.

Time complexity: Same as iterative approach but may be less efficient.

**Finding the LCD**

Iteration: Both GCD and LCM function are implanted using iterative algorithms.

Time complexity: GCD- O(log(min(a,b))); O(1) for finding LCM

Recursion: Same time complexity as the iterative approach, but It may be less efficient due to overhead of function calls.

**Raising a Number to an Exponent**

Iteration: A for loop was used to repeatedly multiply the base by itself ‘e’ times.

Time complexity: O(e), constant work done.

Recursion: The function calls itself ‘e’ times until the base case is reached.

Time complexity: same as iterative approach, may be less efficient due to the overhead of function calls.

**Summary**

Both Iterative and recursion approach achieve the same results, but the iterative approach us more efficient in terms of time complexity. Recursion is easier to implement and understand but it can be less efficient sue to the overhead of function calls.

**How to run the code:**

1. **Press the play button to run the code.**
2. **To edit the numbers, You can either customize the number in the code itself, or when it runs, depending on the program.**