CSE 464

Bhavya Patel - ASU ID: 1225740997

A) Testing Approach:

My approach is to test the application using three methods: valid, boundary, and negative testing. For valid tests, I'd use normal leap years (e.g., 2024) and non-leap years (e.g., 2025) to confirm correct behavior. For boundary cases, I'd check tricky rules like years divisible by 100 but not 400 (1900) and those divisible by 400 (2000), as well as nearby years. Finally, for negative testing, I'd input invalid values such as decimals, or negative numbers to ensure the program handles errors properly without crashing.

· **Valid Cases**: Confirm correct results for typical years, especially those well-known as leap or common years (such as 2020, 2019).

· **Boundary Cases**: Focus on years where the leap year logic's rules change, such as years divisible by 100 and 400 (e.g., 1900, 2000).

· **Invalid Inputs**: Include negative years, zero, and non-numeric values to check input validation (e.g., "hello", -2020, 20.40).

B) Test Cases:

Test	Input	Expected Output	Reasoning
1	2020	Leap Year	Divisible by 4, not by 100: regular leap year
2	1800	Not a Leap Year	Divisible by 100, not by 400: exception
3	1600	Leap Year	Divisible by 400: still a leap year
4	2025	Not a Leap Year	Not divisible by 4: normal common year
5	-400	Invalid Input	Negative year, the system should not accept an input less than 0.