1. Prove that if \( d(n) \) is \( O(f(n)) \) and \( e(n) \) is \( O(g(n)) \) then:

- \( d(n) + e(n) \) is \( O(f(n) + g(n)) \)
- \( d(n) \cdot e(n) \) is \( O(f(n) \cdot g(n)) \)

2. Prove using limits that \( f(n) = 12n^2 + 6n \) is \( o(n^3) \) and \( \omega(n) \). Also try to prove the same without using limits.

3. Consider the algorithm that finds the maximum element in a given array. How will you prove the correctness of your algorithm.

4. Show that the functions \( \sqrt{x^2 + 3} \) and \( (2\sqrt{x} - 2)^2 \) grow at the same rate.

5. Show that \( \log_b f(n) \) is \( \Theta(\log_2 f(n)) \) if \( b > 1 \) is a constant.

6. Show that \( \lceil f(n) \rceil \) is \( O(f(n)) \) if \( f(n) \) is always greater than 1.

7. Show that the summation \( \sum_{i=1}^{n} \lceil \log_2 i \rceil \) is \( O(n \log_2 n) \).

8. Show that the summation \( \sum_{i=1}^{n} \lceil \log_2 i \rceil \) is \( \Omega(n \log_2 n) \).

9. Algorithm A uses \( 10n \log n \) operations, while algorithm B uses \( n^2 \) operations. Determine the value of \( n_0 \) (as small as you can without using calculator) such that A is better than B for \( n \geq n_0 \).

10. Write an algorithm and a program to find the maximum element in an array recursively.