Problem 1: 25 Points
(a) Show that $x^2 + 4x + 17$ is $O(n^3)$, but that $O(n^3)$ is not $O(x^2 + 4x + 17)$.
(b) Show that $\log_{10} n = O(\log_2 n)$ and that $\log_2 n = O(\log_{10} n)$.
Note: For part (a) use the definition of $O()$ notation.
For part (b) recall $\log_a b = \log_c b / \log_c a$.

Problem 2: 25 Points
Show that:
(a) $(n^2 + 8)(n + 1) = O(n^3)$.
(b) $(n \log n + n^2)(n^3 + 2) = O(n^5)$.
(c) $n \log^2 n + n^2 \log n = O(n^2 \log n)$.
(d) $2^n \times 3^n = O(2^{(1+\log_2 3)n})$.
Note: For parts (b) and (c) you may assume that $\log n = O(n)$ is known.
For part (d) recall $x \log_x y = y$ and $\log_a b = \log_c b / \log_c a$.

Problem 3: 25 Points
(a) Argue that the base 2 representation of every positive integer $n$ requires $\lceil \log_2 n \rceil$ bits.
Hint: What is the largest integer that you can represent, in base 2, with $N$ bits?
(b) Below is the pseudocode (from Rosen’s book) that constructs the base 2 representation of a positive integer:

```
procedure base 2 expansion (n: positive integer)
    q := n ;
    k := 0 ;
    while q ≠ 0
        \begin
        a_k := q mod 2 ;
        q := \lceil q/2 \rceil ;
        k := k + 1 ;
        \end
    \{ the base 2 expansion of n is $a_{k-1} \ldots a_1 a_0$ \}

    (i) Show all the values of $a_k$ and $q$ on input $n = 2379$.
    (ii) Argue that the while loop is executed $O(\log n)$ times.
    (iii) If $r(n)$ is the complexity of computing the function $n \mod 2$, $q(n)$ is the complexity of computing the function $n \div 2$, and $add(n)$ is the complexity of adding two $n$-bit numbers, argue that the running time of the while loop is $O(\log n (r(n) + q(n) + add(n)))$.

Problem 4: 25 Points
(a) Where $n$ is a positive integer, $\sum_{i=n}^{2n} (3i + 4) =$?
(b) Where $n$ is a positive integer, $\prod_{i=n}^{2n} (2^i \times 4) =$?
Note: Show your calculations.
Problem 5: 25 Points

In the section of Rosen’s book on Integers and Algorithms, there is a subsection on Algorithms for Integer Operations. The algorithms for integer addition, multiplication and division are presented in pseudocode.

For integer addition and multiplication the pseudocode assumes input in base 2 representation. The arguments immediately following the algorithms explain that the addition algorithm with $n$-bit input in base 2 representation uses $O(n)$ bit operations, while the multiplication algorithm with $n$-bit input in base 2 representation uses $O(n^2)$ bit operations.

For integer division the pseudocode abstracts away from input representation, for the purpose of stressing that the while loop goes through $O(q)$ iterations, where $q$ is the output quotient.

(a) Give the algorithm for integer division, in pseudocode, when the input $a$ and $d$ (the dividend and the divisor) are positive integers in base 2 representation.
(b) Argue that the above algorithm uses $O(q \log a)$ bit operations.
(c) Suppose that the base 2 representations of both $a$ and $d$ contain $n$ or fewer bits. Argue that $O(q \log a) = O(n^2)$. 
(d) Suppose that the base 2 representations of both $a$ and $d$ contain $n$ or fewer bits. Give an algorithm, in pseudocode, which uses $O(n^2)$ bit operations. **Hint:** It is the elementary school division algorithm, for binary input.