These are some review questions to test your understanding of the material. Some of these questions may appear on an exam.

1 Asymptotic Analysis

1.1 Define "Big Oh," "Big Omega," and "Big Theta." Use formal, mathematical definitions.

1.2 Number these functions in ascending "Big Oh" order:

Number	Big Oh
	$O(n^3)$
	$O(n^2 \log n)$
	$O(1)$
	$O(\log^{0.1} n)$
	$O(n^{1.01})$
	$O(n^{2.01})$
	$O(2^n)$
	$O(\log n)$
	$O(n)$
	$O(n \log n)$
	$O(n \log^5 n)$

1.3 Prove:

$$\sum_{i=1}^{n} i = \frac{n(n+1)}{2}$$

1.4 Let $T_1(x) = O(f(x))$ and $T_2(x) = O(g(x))$. Prove $T_1(x) + T_2(x) = O(max(f(x), g(x)))$

1.5 Prove: $O(cf(x)) = O(f(x))$, where c is a positive constant.

1.6 Let $T_1(n) = O(f(n))$ and $T_2(n) = O(g(n))$. Prove $T_1(n)T_2(n) = O(f(n)g(n))$

1.7 Prove $2^{n+1} = O(2^n)$

1.8 Prove: If $T(n)$ is a polynomial of degree x, then $T(n) = O(n^x)$

1.9 Prove $\lg^k n = O(n)$ for any positive constant k. Note that $\lg^k n$ means $(\lg n)^k$

1.10 Prove $n^k = O(a^n)$ for $a > 1$.

1.11 Show the following using one or more of the theorems in questions 4, 5, 6 and 7. Be explicit and mention *each* of the theorems that applies.

1. $n^2 + 2n = O(n^2)$
2. $n^2 - 3n = O(n^2)$
3. $n^2 + 98 = O(n^2)$

1.12 The following C++ function computes the maximum contiguous subsequence sum of elements in the given vector.

```
int maxSubSum2 (const vector<int> & a)
{
  int maxSum = 0;

  for (int i = 0; i < a.size(); i++)
  {
    int thisSum = 0;
    for (int j = i; j < a.size(); j++)
    {
      thisSum += a[j];
      if (thisSum > maxSum)
        maxSum = thisSum;
    }
  }
  return maxSum;
}
```

What is the Big-Oh asymptotic time performance of this function, worst-case, as a function of the vector size N?

1.13 The following C++ function computes $\sum_{i=1}^{N} i^3$

```
int sum (int N)
{
  int partialSum;

  partialSum = 0;
  for (int i = 1; i <= N; i++)
    partialSum += i * i * i;
  return partialSum;
}
```

What is the Big-Oh asymptotic time performance of this function, worst-case, as a function of the argument N?