1. Suppose you have a floating-point number system with base $\beta = 8$, with $t = 5$ digits in each number, with exponents ranging from $e_{\text{min}} = -7$ to $e_{\text{max}} = 6$, a bit for representing the sign, and all non-zero numbers have a mantissa (significand) starting with a non-zero digit and then the decimal point.

(a) What's the largest positive number you can represent in this system?
(b) What's the smallest positive number you can represent in this system?
(c) How many different numbers (positive, negative, zero) can you represent in this system?
(d) For the following arithmetic operations, give examples where the input numbers can be exactly represented in this number system, but the output cannot:
   i. Addition
   ii. Subtraction
   iii. Multiplication

2. What sources of error would you worry about in the following computations, if they were implemented using floating-point arithmetic on a computer? Explain your answers.
   (a) $e^{\pi/3}$
   (b) $\log(x) - \log(x + 1)$ (for arbitrary positive real number $x$).
   (c) $\sum_{i=0}^{n} (2^i/3) + (3/2^i)$ (for arbitrary positive natural number $n$).

3. What is the condition number $(\|f'(x)\| / \|f(x)\|)$ of the following functions, and what does it tell you about calculating the functions when there may be some imprecision in the input?
   (a) $\sqrt{x}$.
   (b) $7x^8$.
   (c) $e^x$. 

March 29, 2010
4. Consider the code for matrix multiplication below, where $A$, $B$, and $C$ are two-dimensional arrays. Assume that $A$, $B$, and $C$ are square $(n \times n)$ arrays, and that each operation that doesn't depend on $n$ costs 1 “step.” Compute an over-estimate of the number of steps required by the algorithm that is bounded above by some constant multiple of $n^3$, and explain your estimate. Compute an underestimate of the number of steps required by the algorithm that is bounded below by some constant multiple of $n^3$, and explain your estimate. What do your computations tell you about this algorithm?

```
MatrixMultiplication(A, B)
1. if (A.cols != B.rows)
2. return null
3. i = 0
4. while (i < A.rows){
5. k = 0
6. while (k < B.cols){
7. C[i][k] = 0
8. j = 0
9. while (j < A.cols){
10. temp = C[i][k]
11. result = A[i][j] \cdot B[j][k]
12. C[i][k] = temp + result
13. j = j + 1
14. k = k + 1
15. i = i + 1
16. return C
```

5. Define $U(n) : \exists k \in \mathbb{N}, n = 7k + 2$, and $V(n) : \exists k \in \mathbb{N}, n = 7k + 4$. Use the formal proof structure from this course to prove that for all natural numbers $n$, $U(n) \Rightarrow V(n^2)$. Is the converse true? Prove (or disprove) the converse, using the formal proof structure from this course.

6. Let $f(n) = 3n + 7n^3$, and let $g(n) = 17 + 34n^2$. Use the formal proof structure from this course to prove that $g \in O(f)$, and that $f \not\in O(g)$.

7. Some graffiti in Robarts Library claims that you won’t finish an undergraduate degree unless you sell your soul. In order to test the claim, the world’s population is divided into four groups:

- People who have sold their souls.
- People who haven’t finished an undergraduate degree.
- People who have not sold their souls.
- People who have finished an undergraduate degree.

Which of the four groups must be questioned, and which can be safely ignored? Explain.
Exam tactics

The exam is nine questions long, marked out of 90, and lasts three hours. It is comprehensive, that is you are responsible for the entire twelve-week semester. Some questions are similar to material you've worked on for assignments, term tests, or tutorials. You always have the option of leaving a question blank or writing “I do not know how to answer this” for 20% of the marks applicable to that portion of the exam. For a formal proof, you will receive roughly half marks if you write a correct outline and do not write any steps that you can’t justify.

Here are some suggestions for the best use of your time.

1. Be sure to be familiar with (although the exam may not be limited to these):
   - Logic, precise logical notation, implication, Venn diagrams.
   - Proof techniques, structured proof format.
   - Big-Oh (big-Omega, big-Theta).
   - Algorithm complexity.
   - Binary numbers, floating-point representation, sources of error.
   - Assignments 1–3, tests 1–3, tutorials 1–6, and their posted solutions.
   - Lecture examples.

2. Read every exam question before starting, since you may find some easier than others.

3. Make sure you understand what you’re being asked to do before you begin writing. When in doubt, ask me or the other invigilator a question and we will try to provide a fair answer.

4. Write the outline of a proof, even if there are steps you can fill in. Indicate which steps you can’t fill in. Specify which things you assume without proof.

5. Use the spaces left on the exam paper as an indication of expected length. There will be some extra blank pages at the end of the exam.