CSE 321: Discrete Structures
Assignment #3
due: Wednesday, October 22

**Reading:** Sections 2,4-2.5, 3.1, 3.3, 3.4 (In 4th edition, these are 2.3-2.4 and 3.3-3.3).

1. Give an example of a function from $\mathcal{N}$ to $\mathcal{N}$ which is
   - one-to-one but not onto
   - onto but not one-to-one
   - both onto and one-to-one (but different from the identity function)
   - neither one-to-one nor onto.

The next two problems use the following definition: Let $g$ be a function from the set $A$ to the set $B$ and let $f$ be a function from the set $B$ to the set $C$. The *composition* of the functions $f$ and $g$, denoted by $f \circ g$, is defined by

$$(f \circ g)(a) = f(g(a)).$$

2. Let $f : \mathcal{R} \rightarrow \mathcal{R}$, where $f(x) = x^3$ and $g : \mathcal{R} \rightarrow \mathcal{R}$, where $g(x) = x - 3$. Give expressions for $f \circ f$, $f \circ g$, $g \circ f$ and $g \circ g$.
3. If $f$ and $f \circ g$ are one-to-one, does it follow that $g$ is one-to-one? Justify your answer.
4. Prove that if $a|b$ and $b|c$, then $a|c$.
5. How many zeros are there at the end of 100! Justify your answer. The function $n!$ is the product of all the integers 1 through $n$. (Hint: Think about the unique factorization of 100! into primes. What about this factorization determines the number of zeros at the end of the decimal representation of 100! ?)
6. Using only your brain, pencil, and paper (e.g., no calculator), compute $23^{25} \bmod 31$. Show your intermediate steps (as proof that you used your brain instead of a calculator). (Hint: If you use the method I demonstrated in lecture, you should never need to compute any product greater than $15 \cdot 15$.)
7. Show that if $a$, $b$ and $m$ are integers such that $m \geq 2$ and $a \equiv b (\mathrm{mod} m)$, then $\gcd(a, m) = gcd(b, m)$.
8. Use Euclid's algorithm to compute the following, showing the values of $x$ and $y$ for each iteration of the algorithm.
   (a) $\gcd(1020, 1173)$
   (b) $\gcd(1019, 1173)$
9. Suppose that you want to compute $\gcd(a, b)$, where $a$ and $b$ each have $n$ digits. The naive algorithm that first finds the prime factorization of $a$ and $b$ uses approximately $10^{n/2}$ integer divisions to do so, by trying all possible divisors up to $\sqrt{a}$ and $\sqrt{b}$, respectively. In contrast, Euclid's algorithm uses approximately $5n$ divisions. Suppose you were running these two algorithms on a computer that could do $10^9$ divisions per second. Put your answers to the following questions into a single $3 \times 2$ table:
   - What is the greatest number $n$ of digits that you could handle by each of the two methods in $10^{-6}$ seconds of computer time?

- What is the greatest number $n$ of digits that you could handle by each of the two methods in $10^{-3}$ seconds of computer time?
- What is the greatest number $n$ of digits that you could handle by each of the two methods in 1 second of computer time?