CSU200 Discrete Structures Professor Fell Integers and Division

Though you probably learned about integers and division back in fourth grade, we need formal definitions and theorems to describe the algorithms we use and to very that they are correct, in general.

If **a** and **b** are integers, $\mathbf{a} \neq \mathbf{0}$, we say **a** *divides* **b** if there is an integer **c** such that **b = ac**. **a** is a *factor* of **b**.

a | b means a divides b.

$a \nmid b$ mean a does not divide b.

Theorem 1: Let a, b, and c be integers, then

1. if a | b and a | c then a | (b + c)
2. if a | b then a | bc for all integers, c
3. if a | b and b | c then a | c.

Proof: Here is a proof of 1. Try to prove the others yourself.
Assume a, b, and c be integers and that a | b and a | c.
From the definition of *divides*, there must be integers m and n such that:

b = ma and c = na.

Then, adding equals to equals, we have

b + c = ma + na.

By the distributive law and commutativity,

b + c = (m + n)a.

By closure of addition, m + n is an integer so, by the definition of *divides*,

a | (b + c). Q.E.D.

Corollary: If a, b, and c are integers such that a | b and a | c then a | (mb + nc) for all integers m and n.

Primes
A positive integer **p** > **1** is called *prime* if the only positive factor of **p** are **1** and **p**.

How can you find prime numbers?
The mathematician, **Eratosthenes** (276-194 BC) invented a prime number sieve, the **Sieve of Eratosthenes**, which, in modified form, is still used in number theory research.

How many primes must you sieve by to find all the prime numbers less than 100?
How many primes must you sieve by to find all the prime numbers less than n, where n is a positive integer?

Fundamental Theorem of Arithmetic: Every positive integer greater than 1 can be written uniquely as a prime or as the product of two or more primes were the prime factors are written in order of non-decreasing size.

Examples:

364 = 2*2*7*13 = 2^2*7*13
7581 = 7*19*57
32769 = 2^{15}
31752 = $2^3 3^4 7^2$

Theorem: There are infinitely many primes.
Can you prove this?

Want to listen to some primes? Try the Prime Number Listening Guide.

Some functions related to division

Back in elementary school, you probably wrote out division problems like this:

$$\begin{array}{r} 4 \\ 7\overline{)29} \end{array} \quad r = 1$$

In this equation, 29 is the *dividend*, 7 is the *divisor*, 4 is the *quotient*, and 1 is the *remainder*. Here is a general statement about division of integers.

The Division "Algorithm": Let ***a*** be an integer and ***b*** a positive integer. Then there are unique integers ***q*** and ***r***, with ***0*** £ ***r*** < ***b***, such that ***a*** = ***bq*** + **r**.

b is the *divisor*, ***a*** is the *dividend*, ***q*** is the *quotient*, and ***r*** is the *remainder*.

Scheme Functions related to the Division Algorithm

procedure: (quotient int1 int2)
returns: the integer quotient of int1 and int2

```
(quotient 45 6) ⇒ 7
(quotient 6.0 2.0) ⇒ 3.0
(quotient 3.0 -2) ⇒ -1.0
```

procedure: (remainder int1 int2)
returns: the integer remainder of int1 and int2

The result of remainder has the same sign as int1.

```
(remainder 16 4) ⇒ 0
(remainder 5 2) ⇒ 1
(remainder -45.0 7) ⇒ -3.0
(remainder 10.0 -3.0) ⇒ 1.0
(remainder -17 -9) ⇒ -8
```

modulo is similar to but not quite the same as remainder.

procedure: (modulo int1 int2)
returns: the integer modulus of int1 and int2

The result of modulo has the same sign as int2.

```
(modulo 16 4) ⇒ 0
(modulo 5 2) ⇒ 1
(modulo -45.0 7) ⇒ 4.0
(modulo 10.0 -3.0) ⇒ -2.0
(modulo -17 -9) ⇒ -8
```

In some computing languages, the functions *quotient* and *modulo* are called *div* and *mod*. Mathematicians write "a *mod* b" instead of "(modulo a b)." We will see more about *mod* or *modulo* when we do modular arithmetic.

Greatest common divisor and Least common multiple

Let a and b be integers, not both 0. The *greatest common divisor* of **a** and **b**, **gcd(a, b)** is the largest integer **d** such that **d|a** and **d|b**.

Examples:

gcd(75, 21) = 3
gcd(52, 81) = 1
gcd(2^2*7*13, 2^3*3^4*7^2) = 2^2*3^0*7*13 What is the rule?
gcd(49831, 825579) = ? We will soon learn a way to solve. this.

Integers **n** and **m** are *relatively prime* if **gcd(m, n) = 1**.

Let a and b positive integers. The *least common multiple* of **a** and **b**, **lcm(a, b)** is the smallest integer divisible by both **a** and **b**.

Examples:

lcm(75, 21) = 7*75 = 25*21 = 525.
lcm(52, 81) = 52*81 = 4212.
lcm(2^2*7*13, 2^3*3^4*7^2) = 2^3*3^4*7^2*13 What is the rule?

Scheme Functions gcd and lcm

procedure: (gcd int ...)
returns: the greatest common divisor of its arguments int ...

The result is always nonnegative, i.e., factors of -1 are ignored. When called with no arguments, gcd returns 0.

```
(gcd) ⇒ 0
(gcd 34) ⇒ 34
(gcd 33.0 15.0) ⇒ 3.0
(gcd 70 -42 28) ⇒ 14
```

procedure: (lcm int ...)
returns: the least common multiple of its arguments int ...

The result is always nonnegative, i.e., common multiples of -1 are ignored. Although lcm should probably return ∞ when called with no arguments, it is defined to return 1. If one or more of the arguments is 0, lcm returns 0.

```
(lcm) ⇒ 1
(lcm 34) ⇒ 34
(lcm 33.0 15.0) ⇒ 165.0
(lcm 70 -42 28) ⇒ 420
(lcm 17.0 0) ⇒ 0
```

Theorem: Let **a** and **b** be positive integers. Then **ab = gcd(a, b)*lcm(a, b)**.

Euclidean Algorithm
How do you find gcd(49831, 825579)? The *Euclidean Algorithm* is a method to compute the gcd of two integers, a and b (not zero). The method is based on the Division Algorithm.

Euclidean Algorithm: If r is the remainder when a is divided by b, i.e. $a = bq + \mathrm{r}$, with *0 £ r* < b, then, gcd(a,b)=gcd(b,r).
Proof:
This follows from the Division Algorithm and the definition of *divides*.

Here is a Scheme implementation of the Euclidean Algorithm from Wikipedia, the free encyclopedia.

```
(define (gcd a b)
     (if (= b 0)
         a
         (gcd b (modulo a b))))
```

For further discussion of the Euclidean algorithm, see the Prime Pages glossary entry. The Visible Euclidean Algorithm is a tool that computes gcd's. Remember that you are supposed to understand the Euclidean Algorithm and will have to perform it by hand on exams.

One of the uses of the Euclidean algorithm is to solve the equation $ax+by = c$. Given a, b, and c, his is solvable (for x and y) whenever the gcd(a,b) divides c. If you keep track of the quotients in the Euclidean algorithm while finding gcd(a,b), you can reverse the steps to find x and y. This method is called the *Extended Euclidean Algorithm*.

Applications
Some encryption methods require finding the gcd of two numbers or finding two numbers whose gcd is 1. RSA is a commonly used Public Key Cryptosystem. The following is from a slide (Fikret Ercal. Ohio State U.) showing requirements for setting up RSA.

Public Key Cryptosystem (RSA)

- p and q are two prime numbers.
- n = pq
- m = (p-1)(q-1)
- a is such that 1 < a < m and gcd(m,a) = 1.
- b is such that (ab) mod m = 1.
- a is computed by generating random positive integers and testing gcd(m,a) = 1 using the extended Euclid's gcd algorithm.
- The extended Euclid's gcd algorithm also computes b when gcd(m,a) = 1.

Adding fractions requires finding the lcm of the denominators.