4-27-2006
What did we talk about last class?
Go ahead and turn in the tests.
You all know that this is, but we will define it properly.

- $c$ is divisible by $b$ iff there exists an integer $n$ such that $bn = c$.

- Prime numbers are numbers divisible by 1 and themselves. Composite numbers are divisible by other values.

- All numbers have a unique prime factorization, the set of primes whose product is equal to the given number.
- If two numbers are divisible by a third number, that third number is called a common divisor.
- The largest number that divides two other numbers is called the greater common divisor (gcd).
- The if gcd(a,b)=1 then a and b are said to be relatively prime.
- Euclid's algorithm provides an efficient way to calculate the gcd.
  - Euclid(a,b) for a>=b
    → if(b==0) a else Euclid(b,a mod b)
- This algorithm scales as the inverse Fibonacci numbers.
An extended form of Euclid's algorithm calculates not only \( d = \gcd(a,b) \), but also gives \( x \) and \( y \) such that \( d = ax + by \). Note that \( x \) and \( y \) can be zero or negative.

- **Extended-Euclid(a,b)**
  - if \( b == 0 \) [\( a,1,0 \) else
  - let \( [d',x',y'] = \text{Extended-Euclid}(b,a \mod b) \)
  - in \( [d',y',x' - ((\text{int})(a/b))y'] \)
Mod defines finite abelian groups.

We want to solve the equation \( ax=b \) (mod \( n \)). This equation either has \( d=\gcd(a,n) \) solutions or zero solutions.

Modular-Linear-Equation-Solver\((a,b,n)\)
- let \([d,x',y']\)=Extended-Euclid\((a,n)\)
- in if\((d|b)\) then
  - let \( x_0=x'(b/d) \) mod \( n \)
  - in the set \( (x_0+i(n/d)) \) mod \( n \) for \( 0\leq i<d \)
- else no solution
We talked about raising numbers to powers in mod space last class and how you can take a mod after each multiply because of closure.

It turns out that \( a^i \pmod{n} \) will form a repeating sequence.

**Modular-Exponentiation(a,b,n)**

- \( c=0, \ d=1 \)
- let \( b_k \) be the bits of \( b \)
- for \( (i=k; \ i>=0; \ --i) \)
  - \( c=2c, \ d=d*d \pmod{n} \)
  - if \( (b_i==1) \) \{ \( c++\); \( d=(d*a) \pmod{n} \); \}
- return \( d \)

\( c \) is not really needed. Just included because \( d=a^c \pmod{n} \)
RSA is a public key system.

Pick two distinct large primes \( p \) and \( q \) and have \( n=pq \).

Pick a small odd integer, \( e \), that is relatively prime to \( (p-1)(q-1) \).

Computer \( d \) such that \( ed=1 \pmod{(p-1)(q-1)} \).

\((e,n)\) is the public key. \((d,n)\) is the private key.

Encode the message \( M \) with \( C=M^e \pmod{n} \).

Decode the message with \( M=C^d \pmod{n} \).

Note \( d \) can only be found if you know \( p \) and \( q \) which require factoring \( n \).
Remember to turn in everything you want to turn in by the 5th.