Domain Pos Real

\[ x^2 \rightarrow x \]

One-to-one \( \times \) onto \( \times \)

\[ f : A \rightarrow B \]

For added security, after we encrypt the data stream, we send it through our Navajo code talker.

\[ \text{\textasciitilde\text{\textasciitilde\textasciitilde}} \text{Is he just using Navajo words for "zero" and "one"?} \]

Whoa, hey, keep your voice down!
### Arithmetic mod 7

\[
\begin{align*}
6 \times_7 4 &= 3 \\
a +_7 b &= (a + b) \mod 7 \\
a \times_7 b &= (a \times b) \mod 7
\end{align*}
\]

#### Table

<table>
<thead>
<tr>
<th>+</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>6</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
</tr>
</tbody>
</table>

#### Table (continued)

<table>
<thead>
<tr>
<th>\times</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>2</td>
<td>4</td>
<td>6</td>
<td>1</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>3</td>
<td>6</td>
<td>2</td>
<td>5</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>4</td>
<td>1</td>
<td>5</td>
<td>2</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>0</td>
<td>5</td>
<td>3</td>
<td>1</td>
<td>6</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>
Let $a$ be an integer and $d$ a positive integer. Then there are unique integers $q$ and $r$, with $0 \leq r < d$, such that $a = d \cdot q + r$.

$$q = a \; \text{div} \; d \quad r = a \; \text{mod} \; d$$

$a = -13 \quad d = 4$
$q = -4 \quad r = 3$

$a = dq \iff d \mid a$

Note: $r \geq 0$ even if $a < 0$. Not quite the same as $a \; \% \; d$. 
Let $a$ and $b$ be integers, and $m$ be a positive integer. We say $a$ is **congruent** to $b$ modulo $m$ if $m$ divides $a - b$. We use the notation $a \equiv b \pmod{m}$ to indicate that $a$ is congruent to $b$ modulo $m$.

\[
a \equiv b \pmod{m} \iff m \mid a-b
\]
modular arithmetic: examples

\[ a \equiv b \mod m \iff a \equiv b + m \mod m \]

A \equiv 0 \pmod{2}
This statement is the same as saying “A is even”; so, any A that is even (including negative even numbers) will work.

1 \equiv 0 \pmod{4}
This statement is false. If we take it mod 1 instead, then the statement is true.

A \equiv -1 \pmod{17} = 16 \mod 17
If A = 17x - 1 = 17(x-1) + 16 for an integer x, then it works.
Note that \((m-1) \mod m\)
\[ \equiv ((m \mod m) + (-1 \mod m)) \mod m \]
\[ \equiv (0 + -1) \mod m \]
\[ \equiv -1 \mod m \]
Theorem: Let \( a \) and \( b \) be integers, and let \( m \) be a positive integer. Then \( a \equiv b \pmod{m} \) if and only if \( a \mod m = b \mod m \).

Proof:

\[
a \equiv b \mod m \iff a \mod m = b \mod m
\]

\[
\Rightarrow \text{ (only if) direction}
\]

\[
m \mid a - b \iff a - b = m \cdot k
\]

\[
\exists k \quad a - b = m \cdot k
\]

\[
a = b + m \cdot k \implies m \cdot b \div m + b \mod m
\]

\[
a \mod m = (b + m \cdot k) \mod m = b \mod m + m \cdot k \mod m = b \mod m
\]

\[
= b \mod m
\]
Theorem: Let $a$ and $b$ be integers, and let $m$ be a positive integer. Then $a \equiv b \pmod{m}$ if and only if $a \mod m = b \mod m$.

Proof: $\Rightarrow$
Suppose that $a \equiv b \pmod{m}$.
By definition: $a \equiv b \pmod{m}$ implies $m | (a - b)$
which by definition implies that $a - b = km$ for some integer $k$.
Therefore $a = b + km$.
Taking both sides modulo $m$ we get
$a \mod m = (b+km) \mod m = b \mod m$
**Theorem:** Let $a$ and $b$ be integers, and let $m$ be a positive integer. Then $a \equiv b \ (\text{mod} \ m)$ if and only if $a \mod m = b \mod m$.

**Proof:**

\[
a \mod m = b \mod m \implies a = b \mod m.
\]

\[
a = (a \div m) \cdot m + a \mod m
\]

\[
b = (b \div m) \cdot m + b \mod m
\]

\[
a - b = m(a \div m - b \div m) + a \mod m - b \mod m
\]

\[
\exists k \quad a - b = mk
\]

\[
ml a - b
\]

\[
a \equiv b \mod m.
\]
**Theorem:** Let $a$ and $b$ be integers, and let $m$ be a positive integer. Then $a \equiv b \pmod{m}$ if and only if $a \mod m = b \mod m$.

**Proof:** \(\Leftarrow\)

Suppose that $a \mod m = b \mod m$.

By the division theorem, $a = mq + (a \mod m)$ and $b = ms + (b \mod m)$ for some integers $q,s$.

$$a - b = (mq + (a \mod m)) - (ms + (b \mod m))$$
$$= m(q - r) + (a \mod m - b \mod m)$$
$$= m(q - r) \text{ since } a \mod m = b \mod m$$

Therefore $m \mid (a-b)$ and so $a \equiv b \pmod{m}$
Let $m$ be a positive integer. If $a \equiv b \pmod{m}$ and $c \equiv d \pmod{m}$, then $a + c \equiv b + d \pmod{m}$. 

$$a \equiv b \pmod{m} \iff m | a - b \iff a - b = m \cdot k$$

$$c \equiv d \pmod{m} \iff m | c - d \iff c - d = m \cdot j$$

$$a = b + c - d = m \cdot (j + k)$$

$$m | a + c - (b + d)$$

$$a + c \equiv b + d \pmod{m}.$$
Let $m$ be a positive integer. If $a \equiv b \pmod{m}$ and $c \equiv d \pmod{m}$, then $a + c \equiv b + d \pmod{m}$.

Suppose $a \equiv b \pmod{m}$ and $c \equiv d \pmod{m}$. Unrolling definitions gives us some $k$ such that $a - b = km$, and some $j$ such that $c - d = jm$.

Adding the equations together gives us $(a + c) - (b + d) = m(k + j)$. Now, re-applying the definition of mod gives us $a + c \equiv b + d \pmod{m}$.
Let $m$ be a positive integer. If $a \equiv b \pmod{m}$ and $c \equiv d \pmod{m}$, then $ac \equiv bd \pmod{m}$.

Suppose $a \equiv b \pmod{m}$ and $c \equiv d \pmod{m}$. Unrolling definitions gives us some $k$ such that $a - b = km$, and some $j$ such that $c - d = jm$.

Then, $a = km + b$ and $c = jm + d$. Multiplying both together gives us

$$ac = (km + b)(jm + d) = k jm^2 + kmd + jmb + bd$$

Rearranging gives us $ac - bd = m(kjm + kd + jb)$. Using the definition of mod gives us $ac \equiv bd \pmod{m}$. 
Let $n$ be an integer.

Prove that $n^2 \equiv 0 \pmod{4}$ or $n^2 \equiv 1 \pmod{4}$

\[ Pf \]

**Case 1:** $n \equiv 0 \mod{2}$

\[ n = 2k \]
\[ n^2 = 4k^2 \]
\[ 4 \mid n^2 \]
\[ n^2 \equiv 0 \mod{4} \]

**Case 2:** $n \equiv 1 \mod{2}$

\[ n = 2k + 1 \]
\[ n^2 = 4k^2 + 4k + 1 = 4(k^2 + k) + 1 \]
\[ n^2 \equiv 1 \mod{4} \]

\[ Pf 2. \]

$n \equiv 0 \mod{4} \implies n^2 \equiv 0^2 \equiv 0 \mod{4}$

$n \equiv 1 \mod{4} \implies n^2 \equiv 1^2 \equiv 1 \mod{4}$

$n \equiv 2 \mod{4} \implies n^2 \equiv 2^2 \equiv 0 \mod{4}$

$n \equiv 3 \mod{4} \implies n^2 \equiv 3^2 \equiv 1 \mod{4}$.
Let $n$ be an integer.
Prove that $n^2 \equiv 0 \pmod{4}$ or $n^2 \equiv 1 \pmod{4}$

**Case 1 (n is even):**
Suppose $n \equiv 0 \pmod{2}$.
Then, $n = 2k$ for some integer $k$.
So, $n^2 = (2k)^2 = 4k^2$.
So, by definition of congruence, $n^2 \equiv 0 \pmod{4}$.

**Case 2 (n is odd):**
Suppose $n \equiv 1 \pmod{2}$.
Then, $n = 2k + 1$ for some integer $k$.
So, $n^2 = (2k + 1)^2 = 4k^2 + 4k + 1 = 4(k^2 + k) + 1$.
So, by definition of congruence, $n^2 \equiv 1 \pmod{4}$. 
Represent integer $x$ as sum of powers of 2:

If $x = \sum_{i=0}^{n-1} b_i 2^i$ where each $b_i \in \{0,1\}$

then representation is $b_{n-1} \cdots b_2 b_1 b_0$

$99 = 64 + 32 + 2 + 1$

$18 = 16 + 2$

For $n = 8$:

$99: \ 0110 \ 0011$

$18: \ 0001 \ 0010$
n-bit signed integers
Suppose $-2^{n-1} < x < 2^{n-1}$
First bit as the sign, n-1 bits for the value

$99 = 64 + 32 + 2 + 1$
$18 = 16 + 2$

For $n = 8$:

99: 0110 0011
-18: 1001 0010

Any problems with this representation?
two's complement representation

n-bit signed integers, first bit will still be the sign bit

Suppose $0 \leq x < 2^{n-1}$,
  $x$ is represented by the binary representation of $x$
Suppose $0 \leq x \leq 2^{n-1}$,
  $-x$ is represented by the binary representation of $2^n - x$

Key property: Two’s complement representation of any number $y$
  is equivalent to $y \mod 2^n$ so arithmetic works mod $2^n$

\[
\begin{align*}
99 &= 64 + 32 + 2 + 1 \\
18 &= 16 + 2
\end{align*}
\]

For $n = 8$:
\[
\begin{align*}
99: & \quad 0110\,0011 \\
-18: & \quad 1110\,1110
\end{align*}
\]
## sign-magnitude vs. two’s complement

<table>
<thead>
<tr>
<th></th>
<th>-7</th>
<th>-6</th>
<th>-5</th>
<th>-4</th>
<th>-3</th>
<th>-2</th>
<th>-1</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1111</td>
<td>1110</td>
<td>1101</td>
<td>1100</td>
<td>1011</td>
<td>1010</td>
<td>1001</td>
<td>0000</td>
<td>0001</td>
<td>0010</td>
<td>0011</td>
<td>0100</td>
<td>0101</td>
<td>0110</td>
<td>0111</td>
</tr>
</tbody>
</table>

### Sign-Magnitude

<table>
<thead>
<tr>
<th></th>
<th>-8</th>
<th>-7</th>
<th>-6</th>
<th>-5</th>
<th>-4</th>
<th>-3</th>
<th>-2</th>
<th>-1</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1000</td>
<td>1001</td>
<td>1010</td>
<td>1011</td>
<td>1100</td>
<td>1101</td>
<td>1110</td>
<td>1111</td>
<td>0000</td>
<td>0001</td>
<td>0010</td>
<td>0011</td>
<td>0100</td>
<td>0101</td>
<td>0110</td>
<td>0111</td>
</tr>
</tbody>
</table>

### Two’s complement
For $0 < x \leq 2^{n-1}$, $-x$ is represented by the binary representation of $2^n - x$.

To compute this: Flip the bits of $x$ then add 1:
- All 1’s string is $2^n - 1$, so
  
  Flip the bits of $x \equiv$ replace $x$ by $2^n - 1 - x$
basic applications of mod

- Hashing
- Pseudo random number generation
- Simple cipher
Scenario:

Map a small number of data values from a large domain \( \{0, 1, \ldots, M - 1\} \) into a small set of locations \( \{0, 1, \ldots, n - 1\} \) so one can quickly check if some value is present.
Scenario:
Map a small number of data values from a large domain \(\{0, 1, ..., M - 1\}\) into a small set of locations \(\{0, 1, ..., n - 1\}\) so one can quickly check if some value is present

- \(\text{hash}(x) = x \mod p\) for \(p\) a prime close to \(n\)
  - or \(\text{hash}(x) = (ax + b) \mod p\)

- Depends on all of the bits of the data
  - helps avoid collisions due to similar values
  - need to manage them if they occur
pseudo-random number generation

Linear Congruential method:

\[ x_{n+1} = (a \times x_n + c) \mod m \]

Choose random \( x_0, a, c, m \) and produce a long sequence of \( x_n \)'s

[good for some applications, really bad for many others]
simple ciphers

- Caesar cipher, $A = 1$, $B = 2$, \ldots
  - HELLO WORLD

- Shift cipher
  - $f(p) = (p + k) \mod 26$
  - $f^{-1}(p) = (p - k) \mod 26$

- More general
  - $f^{-1}(p) = (ap + b) \mod 26$
modular exponentiation mod 7

<table>
<thead>
<tr>
<th>$x$</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>$a$</th>
<th>$a^1$</th>
<th>$a^2$</th>
<th>$a^3$</th>
<th>$a^4$</th>
<th>$a^5$</th>
<th>$a^6$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
modular exponentiation mod 7

<table>
<thead>
<tr>
<th>x</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>4</td>
<td>6</td>
<td>1</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>6</td>
<td>2</td>
<td>5</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>1</td>
<td>5</td>
<td>2</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>3</td>
<td>1</td>
<td>6</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>a</th>
<th>a^1</th>
<th>a^2</th>
<th>a^3</th>
<th>a^4</th>
<th>a^5</th>
<th>a^6</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
modular exponentiation mod 7

<table>
<thead>
<tr>
<th>x</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>4</td>
<td>6</td>
<td>1</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>6</td>
<td>2</td>
<td>5</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>1</td>
<td>5</td>
<td>2</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>3</td>
<td>1</td>
<td>6</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>5</td>
<td>4</td>
<td>3</td>
<td>2</td>
<td>1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>a</th>
<th>a^1</th>
<th>a^2</th>
<th>a^3</th>
<th>a^4</th>
<th>a^5</th>
<th>a^6</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2</td>
<td>4</td>
<td>1</td>
<td>2</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>2</td>
<td>6</td>
<td>4</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>2</td>
<td>1</td>
<td>4</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>4</td>
<td>6</td>
<td>2</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>1</td>
<td>6</td>
<td>1</td>
<td>6</td>
<td>1</td>
</tr>
</tbody>
</table>