There are 10 types of people in the world: those who understand binary and those who don't.
Today's learning goals

- Define the decimal, binary, hexadecimal, and octal expansions of a positive integer.
- Convert between expansions in different bases of a positive integer.
- Define and use the DIV and MOD operators.
- Describe and use algorithms for integer operations based on their expansions.
About you

How many people in this class have you met so far?

A. None.
B. Less than 5.
C. 5-10.
D. 10-15.
E. More than 15.

CENTR 105: AB       CENTR 115: BA

To change your remote frequency
1. Press and hold power button until flashing
2. Enter two-letter code
3. Checkmark / green light indicates success
Algorithms!

From last time

An **algorithm** is a finite sequence of precise instructions for performing a computation or for solving a problem.

… arithmetic

… optimization
Representation

log_2(32)

101

Five
Integer representations

Different contexts call for different representations.

Base 10

Base 2
Bases and algorithms

\[142 \times 17 = 142 = 1 \times 100 + 4 \times 10 + 2 \times 1\]

\[17 = 1 \times 10 + 7 \times 1\]

Decimal expansion: sums of multiples of powers of 10

241
Powers of 10

142 = 1 \times 10^2 + 4 \times 10^1 + 2 \times 10^0

17 = 1 \times 10^1 + 7 \times 10^0

If \( n \) is a positive integer and we express its usual decimal expansion as sums of multiples of powers of 10, then

A. The exponents of powers of 10 in the sum are all positive.
B. There is a largest exponent in the sum.
C. The coefficient of each power of 10 is nonnegative and less than 10.
D. More than one of the above.
E. All of the above.
Powers of \( b \)  

Rosen Theorem 1 Sec 4.1 (p. 246)

For base \( b \) (integer greater than 1) and positive integer \( n \) there is a unique choice of

- \( k \), a nonnegative integer
- \( a_0, a_1, a_2, a_3, \ldots, a_{k-1} \) integers between 0 and \( b-1 \), where
- \( a_{k-1} \neq 0 \) and

\[
\begin{align*}
\text{n} &= a_{k-1}b^{k-1} + \ldots + a_1b + a_0
\end{align*}
\]
Base expansion

Notation: for positive integer \( n \)

Write

\[ n = (a_{k-1} \ldots a_1 a_0)_b \]

when

\[ n = a_{k-1}b^{k-1} + \ldots + a_1b + a_0 \]

Base \( b \) expansion of \( n \)
# Hexadecimal coefficients

<table>
<thead>
<tr>
<th>0</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>9</td>
</tr>
<tr>
<td>2</td>
<td>A</td>
</tr>
<tr>
<td>3</td>
<td>B</td>
</tr>
<tr>
<td>4</td>
<td>C</td>
</tr>
<tr>
<td>5</td>
<td>D</td>
</tr>
<tr>
<td>6</td>
<td>E</td>
</tr>
<tr>
<td>7</td>
<td>F</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
In what base could this expansion be (1401)?

A. Binary (base 2)
B. Octal (base 8)
C. Decimal (base 10)
D. Hexadecimal (base 16)
E. More than one of the above
Base expansion

In what base **could** this expansion be (1401)?

A. Binary (base 2)
B. Octal (base 8)  
   Value: $1*8^3+4*8^2+1=768$
C. Decimal (base 10)  
   Value: $1*10^3+4*10^2+1=1401$
D. Hexadecimal (base 16)  
   Value: $1*16^3+4*16^2+1=5121$
E. More than one of the above
Arithmetic

Addition algorithm in Rosen 4.2 (p. 250)

\[
\begin{align*}
(1401)_{16} + (999)_{16} &= 14096_{10} + 4256_{10} + 1_{10} = 5121_{10} \\
&= (1D9A)_{16}
\end{align*}
\]
Algorithm: constructing base b expansion

**Input** \( n, b \)  

**Output** \( k, \) coefficients in expansion

- English description.
- Pseudocode.
Algorithm: constructing base b expansion

Input n, b  Output k, coefficients in expansion

• English description.
  Find k
  Work down to find \( a_{k-1} \), then \( a_{k-2} \), etc.
Algorithm: constructing base b expansion

Input \( n, b \)  
Output \( k \), coefficients in expansion

- English description.

Find \( k \)

Work down to find \( a_{k-1} \), then \( a_{k-2} \), etc.

What's the length of the base b expansion of \( n \)?

*rephrase as:*

What's the smallest power of \( b \) that is bigger than \( n \)?
Algorithm: constructing base b expansion

Input n, b 
Output k, coefficients in expansion 

English description:
Find k by computing successive powers of b until find smallest k such that 

\[ b^{k-1} \leq n < b^k \]

Initialize value remaining \( v := n \)
For each value of i from 1 to k
    Set \( a_{k-i} \) to be the largest number between 0 and b-1 for which \( a_{k-i} b^{k-i} \leq v \).
    Update \( v := v - a_{k-i} b^{k-i} \)

What's the the base 3 expansion of 17?
A. \((10001)_3\)
B. \((210)_3\)
C. \((111)_3\)
D. \((222)_3\)
E. None of the above.
Algorithm: constructing base b expansion

Input n, b  Output k, coefficients in expansion

English description:

Find k by computing successive powers of b until find smallest k such that

\[ b^{k-1} \leq n < b^k \]

Initialize value remaining \( v := n \)

For each value of \( i \) from 1 to \( k \)

Set \( a_{k-i} \) to be the largest number between 0 and b-1 for which \( a_{k-i} b^{k-i} \leq v \).

Update \( v := v - a_{k-i} b^{k-i} \)

OUR FIELD HAS BEEN STRUGGLING WITH THIS PROBLEM FOR YEARS.

STRUGGLE NO MORE! I’M HERE TO SOLVE IT WITH ALGORITHMS!

SIX MONTHS LATER:
WOW, THIS PROBLEM IS REALLY HARD.
YOU DON’T SAY.
Algorithm: constructing base $b$ expansion \cite{Rosen p. 249}

**Input** $n,b$  \hspace{1cm} **Output** $k$, coefficients in expansion

**Idea:** Find smallest digit first, then next smallest, etc.

…. but how?
Reminder: Divisibility

**Theorem:** For $n$ an integer and $d$ a positive integer, there are unique integers $q$ and $r$ with $0 \leq r < d$ and $n = qd + r$.

**Notation:**
- $q = \frac{n}{d}$ (Quotient when divide $n$ by $b$)
- $r = \mod{n}{d}$ (Remainder when divide $n$ by $b$)

What is $24 \div 5$? What is $15 \mod 5$?

A. 4, 0  
B. 3, 0  
C. 4, 3  
D. 3, 3  
E. I don't know.
When \( k > 1 \)

\[ n = a_{k-1} b^{k-1} + \ldots + a_1 b + a_0 \]
When k > 1

\[ n = a_{k-1}b^{k-1} + \ldots + a_1b + a_0 \]

\[ = b(a_{k-1}b^{k-2} + \ldots + a_1) + a_0 \]

\[ \text{Coeff} \]

\[ a_i \text{ are in } [0, b - 1] \]

\[ q = n \text{ div } d \]

\[ r = n \text{ mod } d \]
Bases and Divisibility

Useful fact:
if \( n = (a_{k-1}...a_0)_b \) then \( bn = (a_{k-1}...a_00)_b \)

Why?

How is this useful?
Algorithm: constructing base b expansion  

Input n,b  
Output k, coefficients in expansion

Idea: Use $n \mod b$ to compute least significant digit. Use $n \div b$ to compute new integer whose expansion we need. Repeat.
Algorithm: constructing base b expansion  

**Input** n, b  

**Output** k, coefficients in expansion  

**Pseudocode:**

```plaintext
procedure base b expansion(n, b : pos ints with b > 1)

1. q := n
2. k := 0
3. while q ≠ 0
4.   ak := q mod b
5.   q := q div b
6.   k := k + 1
7. return (ak−1, ..., a1, a0)
```
Algorithm: constructing base b expansion  
Rosen p. 249

procedure base b expansion\( (n, b : \text{pos ints with } b > 1) \)

1.  \( q := n \)
2.  \( k := 0 \)
3.  while \( q \neq 0 \)
4.   \( a_k := q \mod b \)
5.  \( q := q \div b \)
6.  \( k := k + 1 \)
7.  return \( (a_{k-1}, \ldots, a_1, a_0) \)

\[
\begin{array}{|c|c|c|c|c|}
\hline
n & b & q & k & a_k \\
\hline
17 & 3 & 17 & 0 & 17 \mod 3 = 2 \\
\hline
17 & 3 & 5 & 1 & 17 \div 3 = 5 \quad 5 \mod 3 = 2 \\
\hline
5 & 3 & 1 & 2 & 5 \div 3 = 1 \quad 1 \mod 3 = 1 \\
\hline
1 & 3 & 3 & 3 & 1 \div 3 = 0 \quad \text{return!} \\
\hline
\end{array}
\]

\[
17 = \left( \begin{array}{c} 1 \end{array} \right)_{3}
\]

Definite? Finite? Correct?
**Terminology**

- **Base b expansion**:  
  - Coefficients between 0 and b-1 (inclusive)  
  - Leading coefficient nonzero

- **Fixed-width base b expansion**:  
  - Coefficients between 0 and b-1 (inclusive)  
  - Pad leading coefficients as 0 to match desired width
Representing more?

- Base b expansions can express any **positive integers**
- Fixed width base b expansions can express **nonnegative integers** \([0, b^{k}-1]\)

- What about
  - negative integers?
  - rational numbers?
  - other real numbers?  

  *stay tuned for CSE 30, CSE 140*
Reminders

- Homework 1 due Friday at 11pm
  - Pseudocode and algorithms + number representations

- Office hours available
  - Group
  - One-on-one

There are 10 types of people in the world: those who understand ternary, those who don't, and those who mistake it for binary

(Ternary means base 3)