Q: What is the binary number for 9?

iClicker

A. 10001
B. 01010
C. 01011
D. 01000
E. 01001

CPSC 100

Computational Thinking

Data Representation *Continued*

Instructor: Parsa Rajabi
Department of Computer Science
University of British Columbia

Agenda

- Course Admin
- Learning Goals
- Data Representation *Continued*
  - Binary Numbers [*Review*]
  - Decimal Numbers
  - Hexadecimal

Course Admin

Course Admin

- Parsa (*Instructor*) away for conference [SIGCSE 2025]
  - **Wednesday, Feb 26 → Monday, Mar 3 (inclusive)**
  - **Wednesday, Feb 26** - Kate (TA)
  - **Friday, Feb 28** - Kate/Parsa (TA)
  - **Monday, Mar 3** - Kevin Lindstrom (UBC Librarian)
- **Lab 5** - Number base system in Snap!
  - Due Friday, Feb 28 (extra day provided due midterm week)
- **PC Quiz 4**
  - Released Monday, March 3
  - Due Sunday, March 9

Learning Goals

Learning Goals

After this **today's lecture**, you should be able to:

- Understand the usage of data before and after computers.
- Recognize **hexadecimal numbers** and their role in data representation.
- Count in different **standard number bases** (i.e. 2, 10, 16).
- **Translate** numbers between binary, hexadecimal, and decimal without a calculate

After watching the **take home-video**, you should be able to:

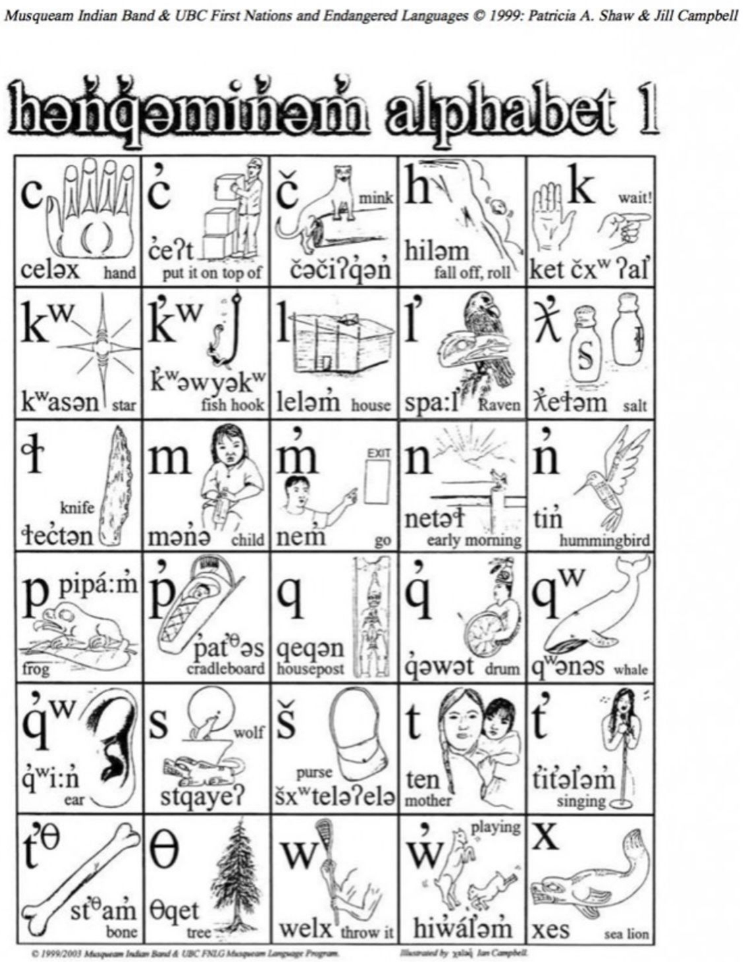
- Recognize the difference between binary and ternary numbers
- Translate numbers between binary, hexadecimal, decimal and **ternary**

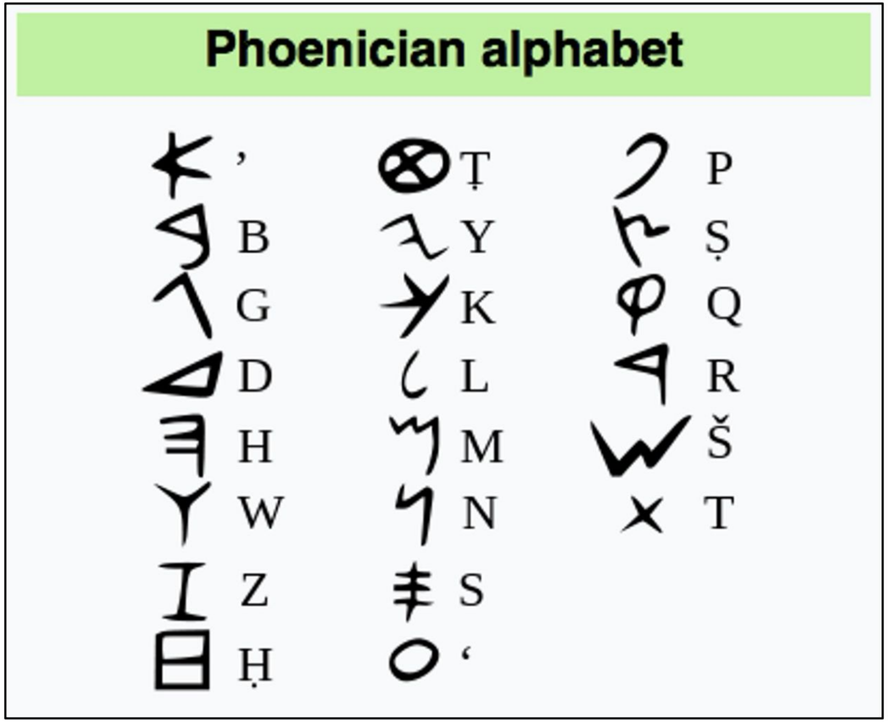
Data Representation Before Computers

Digital Data Before Computers

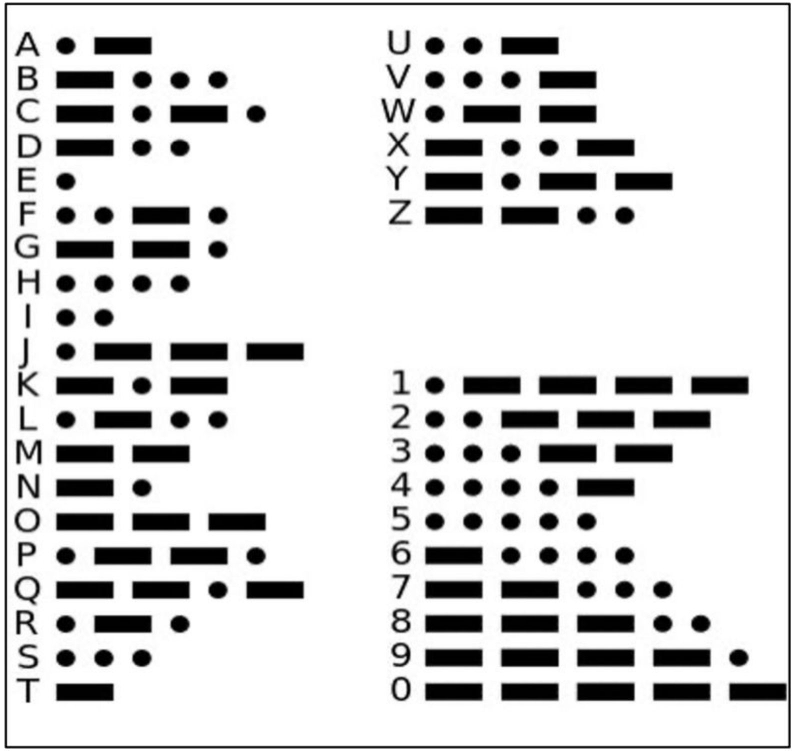
Creative computational thinkers have invented digital data representation systems for millennia, adapting to many different technologies:

Image source: Wikipedia

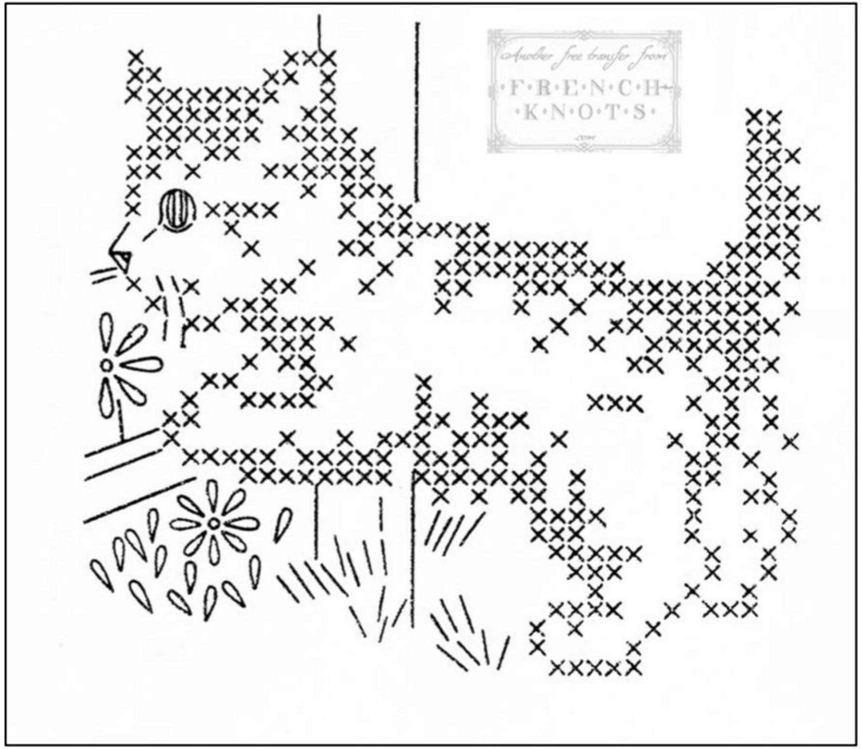




Source: Musqueam Teaching and Learning



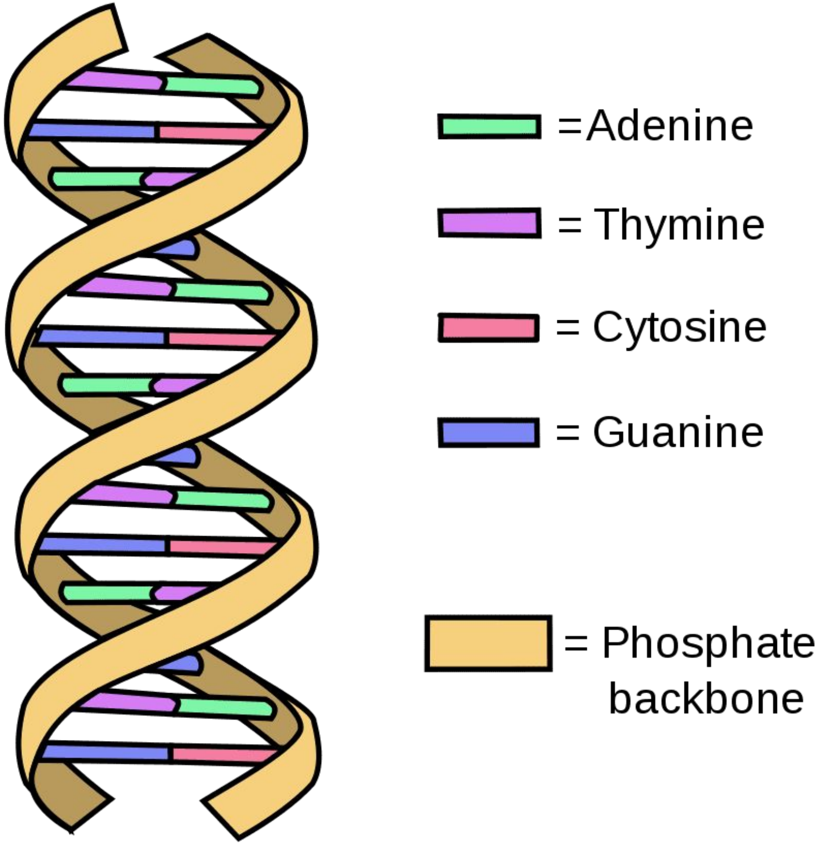
Morse Code



Cross Stitch

Image source: Wikipedia

Data Representation in nature!



DNA

Image source: ashg.org

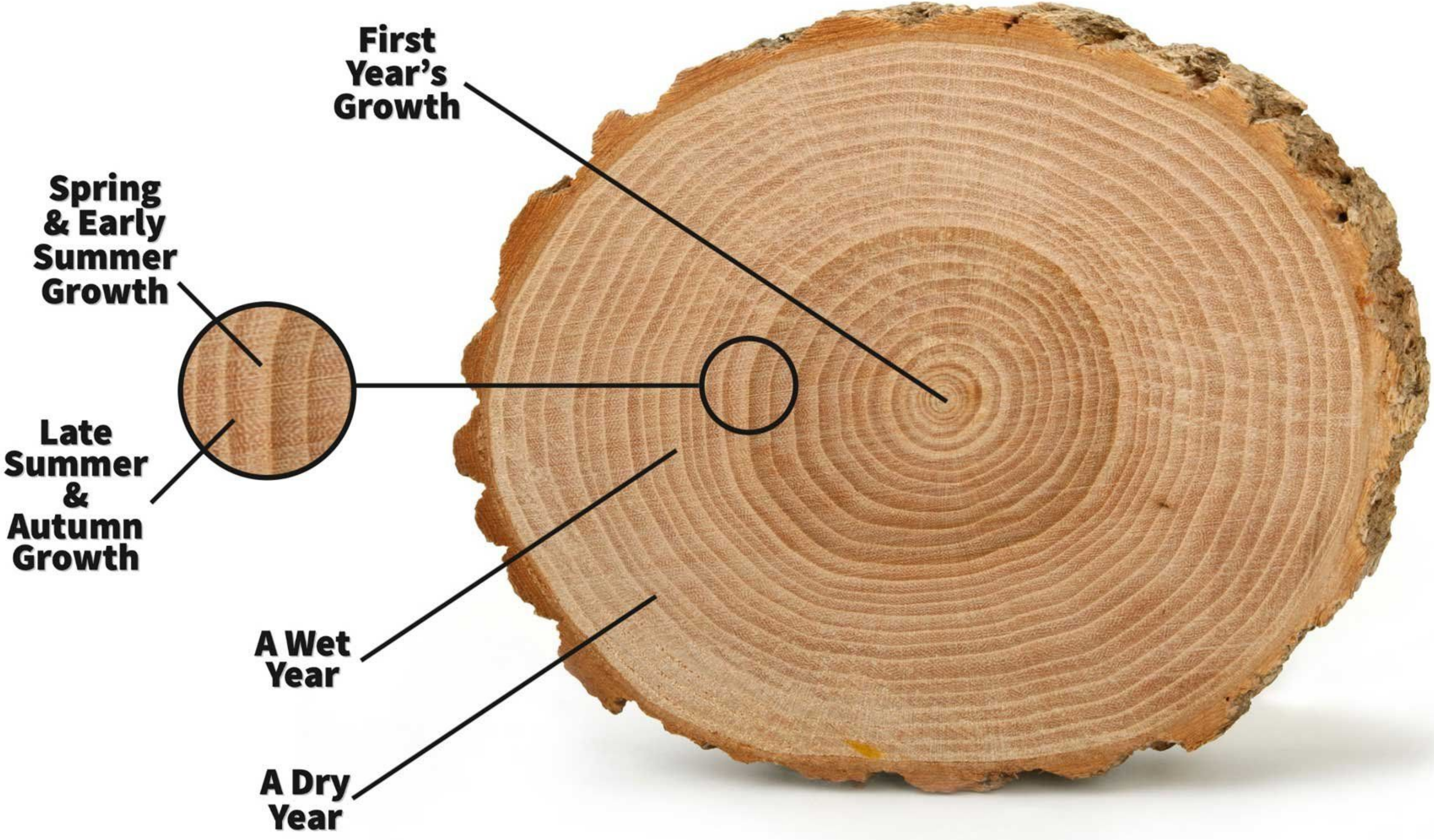


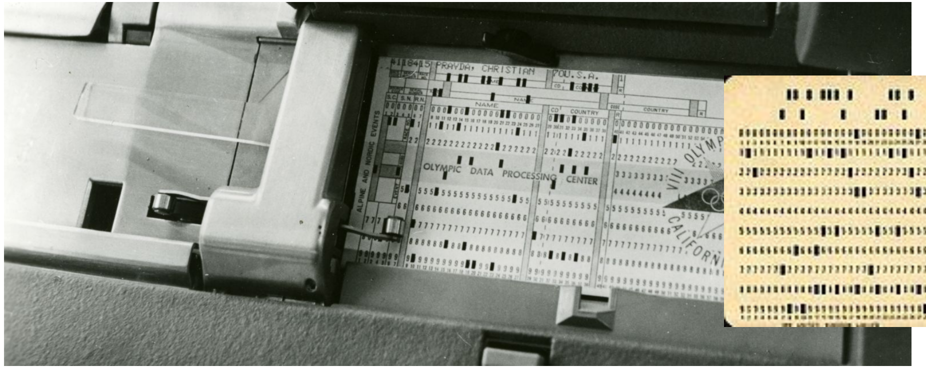
Image source: silvotherapy.co.uk

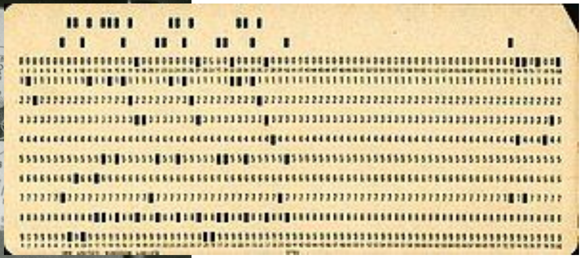
Data Representation with computers

Digital Data with Computers

- Computing technologies represent numbers, text, images, video, and sound using just two symbols: **0's and 1's**

- 0's and 1's are embodied as "high" (on) or "low" (off) signals on various electronic and optical storage media
(e.g., punch cards, vacuum tubes, transistors, DVDs, etc.)





high	low
on	off
true	false
yes	no
+	-
1	0

Image source: Wikipedia

Numbers in a Digital Era

- [Base-10] Decimal Numbers
  - E.g., 1, 2, 3, etc.
- [Base-2] Binary Numbers
  - E.g., 11001, 1001, 1111, etc.
- [Base-16] Hexadecimal Numbers
  - E.g., FF5733, AB4151, etc.
- [Base-3] Ternary Numbers
  - E.g., 12_3, 110_3, 20_3, etc.

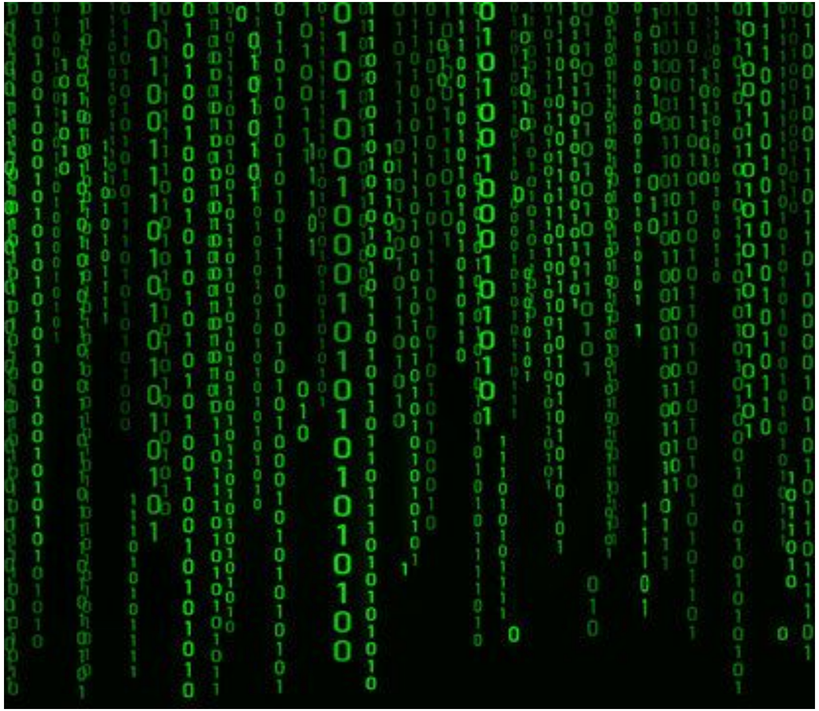


Image source: Adobe

MetaCog Activity - Fill in the Gaps

Decimal	Binary	Hex
00		
01		
02		
03		
04		
05		
06		
07		

Decimal	Binary	Hex
08		
09		
10		
11		
12		
13		
14		
15		

(leading 0’s, shown in gray, are useful for some conversions)

Decimal Numbers

Decimal Numbers

Decimal numbers are base 10.

("decem" is Latin for 10).

In decimal, every column is worth 10 times as much as the previous one

2 4 8

Each digit can be any one of these 10 values: {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}

1s place (10^0)

10s place (10^1)

100s place 100 (10^2)

Counting in Decimal

When counting in decimal:

- Add one to the right most digit
- If that digit has the highest value (9), then add one to the digit in the next column to the left and reset the column you're on to zero.
- If you run out of digits in the column to the left, you repeat the process

- 0
- 1
- 2
- 3
- 4
- 5
- 6
- 7
- 8
- 9

...

99

100

...

10999

11000

Counting in Decimal Example 1

$$\underline{1}\ \ \underline{0}\ \ \underline{0}\ \ +1 \rightarrow \underline{1}\ \ \underline{0}\ \ \underline{1}$$

Add 1 to the
rightmost digit

Counting in Decimal Example 2

1 0 9 +1 → 1 1 0

Add 1 to the rightmost digit. 9 is the highest representation for a number, so we have to change 9 to 0 and increment the number to its left by 1

26

Recall

Binary Numbers

Binary Numbers

Recall

Binary numbers are base 2.

("binarius" is Latin for having two parts).

In binary, every column is worth 2 times as much as the previous one

1 1 0 0 1 ← Binary

- 1s place (2^0)
- 2s place (2^1)
- 4s place (2^2)
- 8s place (2^3)
- 16s place (2^4)

Binary Magic Card Trick

Recall

- Binary numbers only use two digits: 0 and 1, and their place values are based on powers of 2.

Number 9 in Binary:

256	128	64	32	16	8	4	2	1
0	**0**	**0**	**0**	**0**	**1**	**0**	**0**	**1**

9 = 000001001

Counting in Binary

Recall

When counting in binary:

- Add one to the right most digit
- If that digit has the highest value (1), then add one to the digit in the next column to the left and reset the column you're on to zero.
- If you run out of digits in the column to the left, you repeat the process

- 0
- 1
- 10
- 11
- 100
- 101
- …
- 101101
- 101110
- 110001

Counting in Binary Example 1

Recall

1 0 0 +1 → 1 0 1

Add 1 to the rightmost digit

Clicker: Addition in Binary

iClicker

1 0 1 +1 → ? ? ?

What number should go on the right side?
(note: both #s are in binary)

A) 100
B) 101
C) 110
D) 111

MetaCog Activity - Fill in the Gaps

Decimal	Binary	Hex
00	000	
01	001	
02	010	
03		
04	100	
05		
06		
07		

Decimal	Binary	Hex
08		
09	1001	
10		
11	1011	
12		
13	1101	
14		
15		

(leading 0's, shown in gray, are useful for some conversions)

Q: The 8-bit binary representation of 57 is 00111001. What is the 8-bit binary representation of 58?

iClicker

A. 01011110

B. 00111111

C. 00111010

D. 0011100010001

Kibbles and Bits

Kibbles and bits

- A **bit** is short for a **binary digit**.
  - It is one 0 or 1. E.g., one "high" segment on a DVD or an "on" transistor.
- When a computer sees 1000000 (e.g. one "on" part on a DVD followed by six "off" parts on a DVD), it reads this as **binary** and *processes* this as **decimal 64**.
  - 64 and 1000000 are two representations of the same number
- When reading out binary numbers, **we say the digits in turn**
  - For “010” we say "zero one zero" or just "one zero“ (We don't say "ten").
  - Sometimes we write this as **0b010** to make it clear it is a **binary** number.
  - You may also see a subscript 2 after the number (e.g., 010_2).

Convert Binary to Decimal

Recall

0b 1 1 0 0 1

1s place multiply by 1	(2^0):	1
2s place multiply by 2	(2^1):	0
4s place multiply by 4	(2^2):	0
8s place multiply by 8	(2^3):	8
16s place multiply by 16	(2^4):	+ 16
Total in decimal (add them up):		25

Algorithm to Convert Decimal to Binary

- Start with a decimal number *n* (0 to 255)
  - Since 255 is the maximum, we need 8 bits.
- Check the highest power of 2 that fits into *n*:
  - If n < 128, set the 1st (leftmost) bit to 0; otherwise, set it to 1 and subtract 128 from *n* (*n=n-128*)
  - If n < 64, set the 2nd bit to 0; otherwise, set it to 1 and subtract 64 from *n*. (*n=n-64*)
  - If n < 32, set the 3rd bit to 0; otherwise, set it to 1 and subtract 32 from *n*. (*n=n-32*)
  - Continue this process for 16, 8, 4, 2, and finally 1.
- By the end, *n* will be either 0 or 1, which is the last (8th) bit.

Algorithm to Convert 217 to Binary

- n = 217
- Check the highest power of 2 that fits into *n*:
    - ❌ **217** < 128, 1st Bit = 1
        - $n = 217 - 128 \Rightarrow 89$
    - ❌ **89** < 64, 2nd bit = 1
        - $n = 89 - 64 \Rightarrow 25$
    - ✅ **25** < 32, 3rd bit = 0
        - *No subtraction required*
    - ❌ **25** < 16, 4nd bit = 1
        - $n = 25 - 16 \Rightarrow 9$
    - ❌ **9** < 8, 5th Bit = 1
        - $n = 9 - 8 \Rightarrow 1$
    - ✅ **1** < 4, 6th bit = 0
        - *No subtraction required*
    - ✅ **1** < 2, 7th bit = 0
        - *No subtraction required*
    - **Final bit = n = 1 → 8th bit = 1**
- **Final Answer: 11011001**

Take Home Activity

Convert 365_{10} (Decimal) to Binary

Hexadecimal Numbers

Hexadecimal Numbers

- Computers don't understand Decimal (base 10)
- Humans have trouble with Binary (base 2)
  - 1111111110011000111000101010
  - Writing so many 0's and 1's is tedious and error prone
- **Hex digits**, short for **hex**adecimal (base 16)
  - Compromise between humans and computers
  - **Easy to convert between Hex and Binary**

Hexadecimal Numbers

- The digits of the hexadecimal numbering system are
  - 0, 1, 2, 3, 4, 5, 6, 7, 8, 9,   A, B, C, D, E, F
- Because there are 16 digits, they can be represented perfectly by the 16 symbols of 4-bit sequences:
  - The bit sequence **0000** is **hex 0** Bit sequence **0001** is **hex 1**
  - Bit sequence 1111, is hex F
  - Sometimes we use the representation **0x[number]** to show that a number is in **hex**.

MetaCog Activity - Fill in the Gaps

(Take Home)

Decimal	Binary	Hex
00	000	0
01	001	1
02	010	
03		
04	100	
05		
06		6
07		

Decimal	Binary	Hex
08		
09	1001	
10		A
11	1011	
12		
13	1101	
14		
15		F

(leading 0's, shown in gray, are useful for some conversions)

Hexadecimal Numbers to Binary

- Because each hex digit corresponds to a 4-binary sequence, it's easy to translate between hex and binary
  - Hex → binary: replace each hexadecimal digit by the four corresponding binary digits in the conversion table

Hex:	F	A	B	4
Binary:	1111	1010	1011	0100

Binary to Hexadecimal

- Binary → hex:
  - Put extra 0's at the left of the binary number as necessary so that the total number of digits is a multiple of 4
  - then reverse the hex → binary conversion process

Binary:	0010	1011	1010	1101
Hex:	2	B	A	D

MetaCog Activity - Fill in the Gaps
(Take Home)

Decimal	Binary	Hex
00	000	0
01	001	1
02	010	2
03	011	3
04	100	4
05	101	5
06	110	6
07	111	7

Decimal	Binary	Hex
08	1000	8
09	1001	9
10	1010	A
11	1011	B
12	1100	C
13	1101	D
14	1110	E
15	1111	F

(leading 0's, shown in gray, are useful for some conversions)

Q: Translate 0xAFF1 to binary

iClicker

A. 1111 1101 1101 0001

B. 1010 1101 1101 0001

C. 0001 1111 1111 1010

D. 1111 1010 1111 0001

E. 1010 1111 1111 0001

Take Home Activity

Convert 1100 1010 1111 1110 to hexadecimal

Convert 0xA19C to decimal

Convert 48 to hexadecimal

Take-Home Video

(watch before lab)

https://youtu.be/CBYhwcn4WSI?si=fITtLe_y5ZyqDQTA&t=57

Wrap up

Wrap Up

- Parsa (*Instructor*) away for conference [SIGCSE 2025]
  - **Wednesday, Feb 26 → Monday, Mar 3 (inclusive)**
  - **Wednesday, Feb 26** - Kate (TA)
  - **Friday, Feb 28** - Kate/Parsa (TA)
  - **Monday, Mar 3** - Kevin (UBC Librarian)
- **Lab 5** - Number base system in Snap!
  - Due Friday, Feb 28 (extra day provided due midterm week)
- **PC Quiz 4**
  - Released Monday, March 3
  - Due Sunday, March 9