


ANNOUNCEMENTS


- Labs start this week
 - Please attend your scheduled lab session
- Monday: In-class activity
 - Was not a quiz. I did not expect that you would know the answers without looking them up, which is why I mentioned the need to bring materials for reference.
- Wednesday: In-class activity
- Friday: Binary number conversions – representing text.



MONDAY: IN-LECTURE ACTIVITY

- Questions about the history of computing and important computing people.
 - If you were not here on Monday, you can complete the questions and turn in the form by the end of lecture, Friday, September 20 to earn some credit on the assignment.
- 

WEDNESDAY: IN-LECTURE ACTIVITY

- Fill out green half sheet as we are working through the activity.
 - Turn in sheet at end of lecture to earn credit for activity.
 - If you were not in lecture Wednesday, complete the worksheet on binary numbers to earn some credit for the activity.
- 

PART 1 - MAGIC

- Need 3 volunteers to assist



2 3 6 7 10 11 14 15	16 17 18 19 20 21 22 23
18 19 22 23 26 27 30 31	24 25 26 27 28 29 30 31
34 35 38 39 42 43 46 47	48 49 50 51 52 53 54 55
50 51 54 55 58 59 62 63	56 57 58 59 60 61 62 63
8 9 10 11 12 13 14 15	1 3 5 7 9 11 13 15
24 25 26 27 28 29 30 31	17 19 21 23 25 27 29 31
40 41 42 43 44 45 46 47	33 35 37 39 41 43 45 47
56 57 58 59 60 61 62 63	49 51 53 55 57 59 61 63
4 5 6 7 12 13 14 15	32 33 34 35 36 37 38 39
20 21 22 23 28 29 30 31	40 41 42 43 44 45 46 47
36 37 38 39 44 45 46 47	48 49 50 51 52 53 54 55
52 53 54 55 60 61 62 63	56 57 58 59 60 61 62 63



MAGIC? REALLY? (PART 2)

- Well it is a trick – one that can be solved if you understand binary encodings.



PART 3

- Need 5 volunteers to hold up sheets of paper

9 → 01001

27 → 11011



PART 4

- Worksheet: complete the Working with Binary page (can work with others if you'd like)
- Do not need to do Extra for Experts part or other side



ANSWERS TO WORKSHEET

- 10101 is the number 21
- 11111 is the number 31

- Answers to the coded numbers

9	10
5	13
0	17
2	20
0	31



MORE ABOUT BINARY

- If you used each of your fingers to represent a “place” in a binary number, how far could you count?



BASE 2 VS. BASE 10

Base 2

- Two digits used (0, 1)
- Interpretation of number based on increasing powers of 2

Base 10

- Ten digits used (0, 1, 2, 3, 4, 5, 6, 7, 8, 9)
- Interpretation of number based on increasing powers of 10



429

"four hundred twenty-nine"

$$4 \text{ hundreds} = 4 \times 100 = 400$$

$$2 \text{ tens} = 2 \times 10 = 20$$

$$9 \text{ ones} = 9 \times 1 = \underline{+ 9}$$

$$429$$

Algebra:

Any number raised to the zero power is 1.

$$\text{So } 10^0 = 1 \quad 2^0 = 1 \quad 4^0 = 1$$

 101001 ← base 2 number

$$1 \times 2^5 = 1 \times 32 = 32$$

$$0 \times 2^4 = 0 \times 16 = 0$$

$$1 \times 2^3 = 1 \times 8 = 8$$

$$0 \times 2^2 = 0 \times 4 = 0$$

$$0 \times 2^1 = 0 \times 2 = 0$$

$$1 \times 2^0 = 1 \times 1 = \underline{1}$$


(41)

ADDING ZEROS TO THE LEFT SIDE OF NUMBERS

- Adding zeros to the “front” of numbers in base 10 or base 2 does not change the interpretation of the value.


28 *base 10*
 0000028

 101 *base 2*
 00000101



ADDING ZEROS TO THE RIGHT SIDE OF NUMBERS

- Adding zeros to the “back” of numbers does change their interpretation.

- In base 10: *multiplies by 10*
 - $30 \rightarrow 300$
 - In base 2: *multiplies by 2*
 - $1001 \rightarrow 10010$
- 

BINARY IN COMPUTERS

- Each card/place represents a bit (binary digit)
- What is a bit used for?



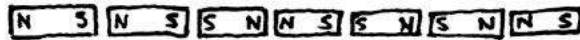
USE OF BINARY ENCODING

- On/off
 - Like a switch
 - Or a transistor
 - Or wire plugged in or not
- High tones and low tones
 - Telephone and radio lines
 - These are the sounds that modems used to make when they connected



BINARY ENCODING

- Magnetic Disks (direction of magnetic field tells us if it's a 0 or 1)
 - Hard drives



- Optical storage (part of the disc does or does not reflect light)
 - CD, CD-ROM, DVD

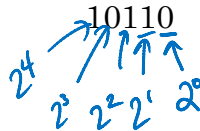


BITS AND BYTES

- 1 bit can not tell us much – we can't store much in it, so we group bits together
- 8 bits = 1 byte
- 1024 bytes = 1 kilobyte (KB)
- 1024 kilobytes = 1 megabyte (MB)
- 1024 megabytes = 1 gigabyte (GB)
- 1024 gigabytes = 1 terabyte (TB)

ALGORITHM FOR CONVERTING FROM BINARY TO DECIMAL

- Recall
 - Binary = Base 2
 - Decimal = Base 10
- Algorithm
 - Input is a number in base 2 (binary)
- STEP 1: Label each digit with the corresponding power of 2



ALGORITHM FOR CONVERTING FROM BINARY TO DECIMAL

- STEP 2: Write out arithmetic expression multiplying the binary digit (0 or 1) with the corresponding power of 2
- STEP 3: Figure out the powers of 2 and substitute into the expression (Hint: It's sometimes easier to do that right to left. Remember anything raised to the zero power is 1.)

$$1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 0 \times 2^0$$

$$(1 \times 16) + (0 \times 8) + (1 \times 4) + (1 \times 2) + (0 \times 1)$$

ALGORITHM FOR CONVERTING FROM BINARY TO DECIMAL

- STEP 4: Do the multiplication

$$16 + 0 + 4 + 2 + 0$$

- STEP 5: Do the addition

$$22$$



ON YOUR OWN

- Convert the binary number 101111 to its decimal equivalent.

$$543210$$

$$1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$$

$$1 \times 32 + 0 \times 16 + 1 \times 8 + 1 \times 4 + 1 \times 2 + 1 \times 1$$

$$32 + 0 + 8 + 4 + 2 + 1$$

$$47$$



CONVERSION FROM DECIMAL TO BINARY (THE OTHER WAY)

- Will demonstrate two different methods for doing this conversion, you only need to learn one – find the one that is easier for you to understand and replicate on an exam.



ALGORITHM FOR CONVERTING FROM DECIMAL TO BINARY #1

- This algorithm relies on division
- Input: A number in Base 10 (decimal)
 - For our first example, let us use the number 49
- STEP 1: Divide 2 into 49 (write it to the far right of the piece of paper you are working on)

$$\begin{array}{r} 24 \text{ R } 1 \\ 2 \overline{) 49} \\ \underline{-48} \\ 1 \end{array}$$



MATH INTERLUDE

- Review of some terms in regards to division
- The answer to a division is called a quotient.
- There is a remainder for the division greater than zero if the division is not an even division. There is a remainder of zero if the division is an even division.



ALGORITHM FOR CONVERTING FROM DECIMAL TO BINARY #1

- STEP 2: Keep dividing 2 into the quotient until the quotient is zero (0). Move each division to the left of the previous one on the paper you are working on.

$$\begin{array}{r}
 0R1 \\
 2 \overline{)1} \\
 \underline{-0} \\
 1
 \end{array}
 \quad
 \begin{array}{r}
 1R1 \\
 2 \overline{)3} \\
 \underline{-2} \\
 1
 \end{array}
 \quad
 \begin{array}{r}
 3R0 \\
 2 \overline{)6} \\
 \underline{-6} \\
 0
 \end{array}
 \quad
 \begin{array}{r}
 6R0 \\
 2 \overline{)12} \\
 \underline{-12} \\
 0
 \end{array}
 \quad
 \begin{array}{r}
 12R0 \\
 2 \overline{)24} \\
 \underline{-24} \\
 0
 \end{array}
 \quad
 \begin{array}{r}
 24R1 \\
 2 \overline{)49} \\
 \underline{-48} \\
 1
 \end{array}$$



ALGORITHM FOR CONVERTING FROM DECIMAL TO BINARY #1

- Write down the remainders from left to right as they appear on your paper. This is the binary version of the number.

110001



ALGORITHM FOR CONVERTING FROM DECIMAL TO BINARY #2

- This algorithm uses the powers of two to help you come up with the binary number.
- Input: A number in Base 10 (decimal)
 - For our first example, let us use the number 49
- STEP 1: Make a list of all the powers of two that are less than the number you are trying to convert.

1 2 4 8 16 32 ~~64~~



ALGORITHM FOR CONVERTING FROM DECIMAL TO BINARY #2

- STEP 2: Subtract from the number, the largest number from the list of the powers of two.

$$\begin{array}{r} 49 \\ -32 \\ \hline 17 \end{array}$$

- STEP 3: Find the next largest power of two that you can subtract from the answer you got from step 2 without making the answer negative.

$$\begin{array}{r} 17 \\ -16 \\ \hline 1 \end{array}$$



ALGORITHM FOR CONVERTING FROM DECIMAL TO BINARY #2

- STEP 4: Repeat this process of subtraction as described in step 3 until the answer is 0.

$$\begin{array}{r} 1 \\ -1 \\ \hline 0 \end{array}$$

- STEP 5: Then, construct the binary number based upon the numbers you subtracted.

$$\frac{1}{32} \quad \frac{1}{16} \quad \frac{0}{8} \quad \frac{0}{4} \quad \frac{0}{2} \quad \frac{1}{1}$$



MORE PRACTICE

- Method #1 – convert the number 56

$$2 \overline{)11}^{0R1} \quad 2 \overline{)13}^{1R1} \quad 2 \overline{)17}^{3R1} \quad 2 \overline{)14}^{7R0} \quad 2 \overline{)28}^{14R0} \quad 2 \overline{)56}^{28R0}$$

111000



MORE PRACTICE


- Method #2 – Convert the number 89

$$\begin{array}{r} 89 \\ -64 \\ \hline 25 \\ -16 \\ \hline 9 \\ -8 \\ \hline 1 \\ -1 \\ \hline 0 \end{array}$$


1011001



MORE PRACTICE PROBLEMS (ON YOUR OWN)

- Convert the following binary numbers to their decimal equivalents
 - 11011
 - 100010
 - 1011100
 - 11010
 - 111001
 - 1000011
- 

MORE PRACTICE PROBLEMS (ON YOUR OWN)

- Convert the following decimal numbers to their binary equivalents (use these problems to determine which method you like better)
 - 23
 - 95
 - 16
 - 43
 - 19
 - 58
 - 79
- 

ENCODING TEXT

- American Standard Code for Information Interchange (ASCII)

- Takes each letter and assigns it a unique 8-bit code

QUESTION: HOW MANY DIFFERENT CHARACTERS CAN BE ENCODED USING 8 BITS?

1-bit → 2 values
2-bits → 4 values
3-bits → 8 values
4-bits → 16 values
5 32
6 64
7 128
8 256

PROBLEMS WITH ASCII?

- Can represent 256 characters
- Is that enough for even some of the world's languages?

