

Numeration systems

This worksheet and all related files are licensed under the Creative Commons Attribution License, version 1.0. To view a copy of this license, visit <http://creativecommons.org/licenses/by/1.0/>, or send a letter to Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA. The terms and conditions of this license allow for free copying, distribution, and/or modification of all licensed works by the general public.

Resources and methods for learning about these subjects (list a few here, in preparation for your research):

Question 1

The numeration system we use in our daily lives is called *base ten*, also called *decimal* or *denary*. What, exactly, does "base ten" mean?

Given the following base-ten number, identify which digits occupy the "one's place," "ten's place," "hundred's place," and "thousand's place," respectively:

5,183

file 01195

Answer 1

"Base ten" means that numbers are represented by combinations of symbols (ciphers), of which there are only ten (0, 1, 2, 3, 4, 5, 6, 7, 8, and 9).

Analyzing the number 5,183:

- One's place: **3**
- Ten's place: **8**
- Hundred's place: **1**
- Thousand's place: **5**

Notes 1

The "base" of our numeration system is something people usually don't think much about – it is simply taken for granted. The purpose of this question is to help students realize what numerical symbols actually mean, in preparation for understanding other systems of numeration.

Question 2

Observe the following sequence of numbers:

00
01
02
03
04
05
06
07
08
09
10
11
12
13
14
15
16
17
18
19
20
21

.
.
.

What pattern(s) do you notice in the digits, as we count upward from 0 to 21 (and beyond)? This may seem like a very simplistic question (and it is!), but it is important to recognize any inherent patterns so that you may understand counting sequences in numeration systems with bases other than ten.

file 01196

Answer 2

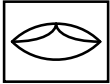
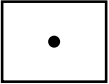
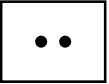
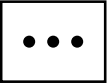
















Note the repeating sequence of digits in the one's place, and the developing pattern of incrementation in the ten's place. How many numbers may be counted without repeating digits in the one's place?

Notes 2

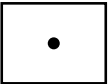
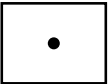
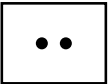
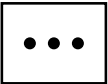
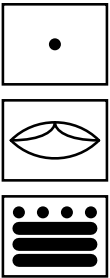
Some students will think this question to be ridiculously simple, but there are some important lessons to be learned here. The greatest problem students face while learning binary, octal, and hexadecimal numeration systems is the overpowering familiarity with base-ten numeration. We are *so* accustomed to base-ten that we don't bother to recognize its basic properties, and typically believe that this is the only way numbers may be written!

Question 3

The ancient Mayans used a *vigesimal*, or base-twenty, numeration system in their mathematics. Each "digit" was actually a composite of dots and/or lines, as such:






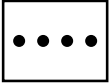


				
Zero	One	Two	Three	Four
				
Five	Six	Seven	Eight	Nine
				
Ten	Eleven	Twelve	Thirteen	Fourteen
				
Fifteen	Sixteen	Seventeen	Eighteen	Nineteen

To represent numbers larger than twenty, the Mayans combined multiple "digits" the same way we do to represent numbers larger than ten. For example:

				
Twenty	Twenty six	Forty eight	Seventy one	Four hundred nineteen

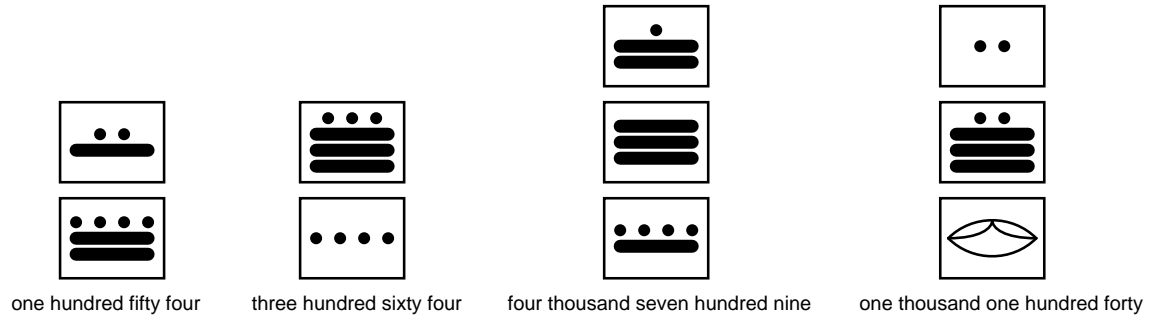
Based on the examples shown above, determine the place-weighting of each "digit" in the vigesimal numeration system. For example, in our denary, or base-ten, system, we have a one's place, a ten's place, a hundred's place, and so on, each successive "place" having ten times the "weight" of the place before it. What are the values of the respective "places" in the Mayan system?

Also, determine the values of these Mayan numbers:

			
			
?	?	?	?

Answer 3

Place weights: one's, twenty's, four hundred's, eight thousand's, . . .



Notes 3

Although no one counts in "Mayan" anymore, this question is still relevant because it gets students thinking outside their accustomed numeration system. Also, it has some cultural value, showing them that not everyone in the world counts the same way they do!

Question 4

Digital computers use a numeration system with a base of *two*, rather than a base of *ten* as we are accustomed to using. It is much easier to engineer circuitry that counts in "binary" than it is to design circuits that count in any other base system. Based on what you know of numeration systems, answer the following questions:

- How many different symbols (ciphers) are there in the binary numeration system?
- What are the different place-weight values in the binary system?
- How would you represent the number "seventeen" in binary?
- In our base-ten (denary) numeration system, each character is called a "digit." What is each character called in the binary numeration system?

file 01198

Answer 4

There are only two valid ciphers in the binary system: 0 and 1. Each successive place carries twice the "weight" of the one before it. Seventeen = 10001 (in binary). Each character in the binary system is called a "bit" rather than a "digit".

Notes 4

Ask your students to hypothesize why binary is used in digital circuitry rather than base-ten (denary) numeration. I've revealed that it's easier to build circuits representing quantities in binary than any other numerical base, but why?

Question 5

If a digital meter display has four digits, it can represent any number from 0000 to 9999. This constitutes ten thousand unique numbers representable by the display. How many unique numbers could be represented by five digits? By six digits?

If an ancient Mayan ledger had spaces for writing numbers with three "digits" each, how many unique numbers could be represented in each space? What if the spaces were expanded to hold four "digits" each?

If a digital circuit has four *bits*, how many unique binary numbers can it represent? If we expanded its capabilities to eight bits, how many unique numbers could be represented by the circuit?

After answering these questions, do you see any mathematical pattern relating the number of "places" in a numeration system and the number of unique quantities that may be represented, given the "base" value ("radix") of the numeration system? Write a mathematical expression that solves for the number of unique quantities representable, given the "base" of the system and the number of "places".

file 01199

Answer 5

Five denary digits: one hundred thousand unique numbers. Six denary digits: one million unique numbers.

Three vigesimal "digits": eight thousand unique numbers. Four vigesimal "digits": one hundred sixty thousand unique numbers.

Four binary bits: sixteen unique numbers. Eight binary bits: two hundred fifty six unique numbers.

$$\text{Unique numbers} = b^n$$

Where,

b = Radix of numeration system.

n = Number of "places" given.

Notes 5

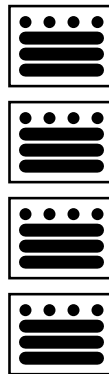
Ask your students which numeration system of the three given is more efficient in representing quantities with the smallest number of places. Then, ask them to explain *why* this is.

Question 6

What is the largest number that can be counted to, in a base-ten system with six digits? How about in a base-twenty (vigesimal) system with four places? How about in a base-two (binary) system with ten bits? [file 01200](#)

Answer 6

999999_{10} = Nine hundred ninety nine thousand, nine hundred ninety nine.



One hundred fifty nine thousand nine hundred ninety nine

111111111_2 = one thousand twenty three.

Notes 6

Ask your students to explain why subscripts are desirable when writing numbers in binary, or in any other "base" numeration system for that matter.

Question 7

How many binary bits are needed to count up to the number one million three hundred thousand seven hundred sixty two? Try to answer this question without converting this quantity into binary form, and then explain the mathematical procedure!

file 01228

Answer 7

Twenty one bits.

Hint: the answer consists in solving this equation:

$$1300762 = 2^n$$

Where,

n = The number of binary bits necessary to count up to 1,300,762.

Notes 7

This question is a test of whether or not students know how to algebraically solve for variable exponents. I'm not going to suggest how this is done (lest I rob a student of the learning experience that comes from researching the answer on their own), but I will say that it is a *very* useful algebraic technique, once mastered.

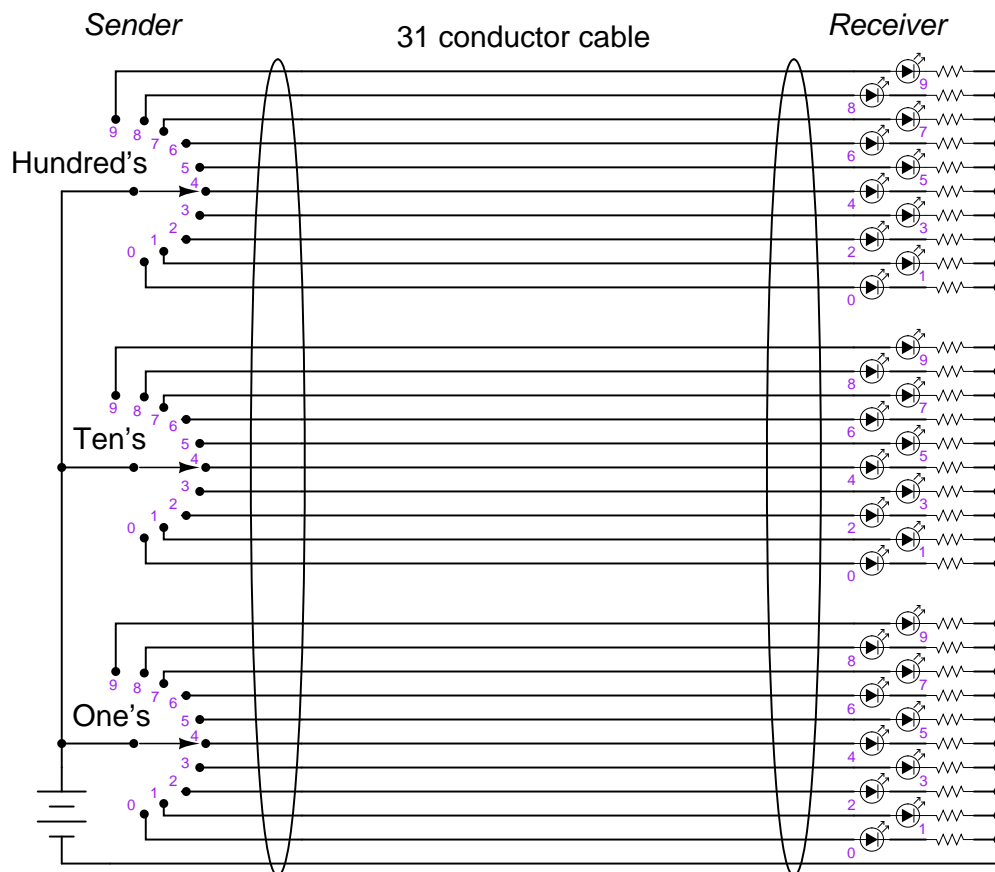
If students still haven't found a solution after doing their research, suggest they try to solve a simpler problem:

How many *decimal digits* are needed to count up to the number one million three hundred thousand seven hundred sixty two?

This question is trivial to answer (7 decimal digits), since we're all familiar with decimal numeration. However, the real learning takes place when students write a mathematical expression for solving this problem, similar to the one written in the answer for the binary problem. Once they have that expression written, ask them what algebraic techniques could be used to solve for the exponent's value.

Question 8

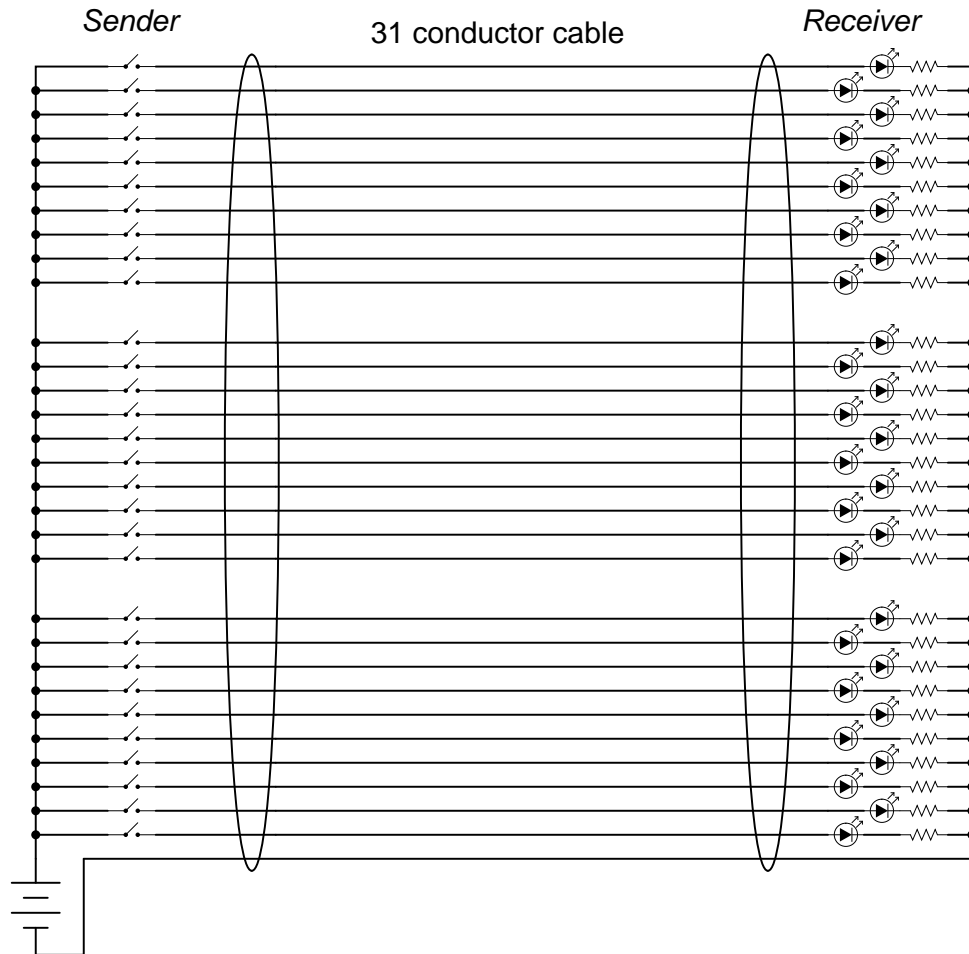
The circuit shown in this diagram is used to transmit a numerical value from one location to another, by means of switches and lights:



Given the switches and lights shown, any whole number between 0 and 999 may be transmitted from the switch location to the light location.

In fact, the arrangement shown here is not too different from an obsolete design of electronic base-ten indicators known as *Nixie tube* displays, where each digit was represented by a neon-filled glass tube in which one of ten distinct electrodes (each in the shape of a digit, 0-9) could be energized, providing glowing numerals for a person to view.

However, the system shown in the above diagram is somewhat wasteful of wiring. If we were to use the same thirty-one conductor cable, we could represent a much broader range of numbers if each conductor represented a distinct binary bit, and we used binary rather than base-ten for the numeration system:



How many unique numbers are representable in this simple communications system? Also, what is the greatest individual number which may be sent from the "Sender" location to the "Receiver" location?
file 01201

Answer 8

In this system, with a "width" of 30 bits, we are able to represent one billion, seventy three million, seven hundred forty one thousand, eight hundred twenty four unique numbers. The greatest individual number which may be communicated is one less than this total.

Notes 8

The purpose of this question is to allow students to consider an electric circuit that communicates digital quantities from one location to another, rather than abstractly discuss numeration systems. It also provides a more practical context in which to understand maximum count in a numeration system.

Question 9

Explain why binary is a natural numeration system for expressing numbers in electronic circuits. Why not decimal or some other base of numeration?

How do you suspect binary numbers may be *stored* in electronic systems, for future retrieval? What advantages are there to the use of binary numeration in storage systems?

file 01203

Answer 9

The two states of an on/off circuit are equivalent to 0 and 1 ciphers in binary. Any medium in which on/off states may be physically represented is applicable to storing binary numbers. Optical disks (CD-ROM, DVD) are an excellent example of this, with laser-burned "pits" representing binary bits.

Notes 9

Ask your students to think of different media or physical quantities which may represent binary information, especially those related to electric/electronic circuits.

Question 10

Convert the following numbers from binary (base-two) to decimal (base-ten):

- $10_2 =$
- $1010_2 =$
- $10011_2 =$
- $11100_2 =$
- $10111_2 =$
- $101011_2 =$
- $11100110_2 =$
- $10001101011_2 =$

Describe a general, step-by-step procedure for converting binary numbers into decimal numbers.
file 01202

Answer 10

- $10_2 = 2_{10}$
- $1010_2 = 10_{10}$
- $10011_2 = 19_{10}$
- $11100_2 = 28_{10}$
- $10111_2 = 23_{10}$
- $101011_2 = 43_{10}$
- $11100110_2 = 230_{10}$
- $10001101011_2 = 1131_{10}$

Notes 10

The most important part of this question, of course, is the *method* of translating binary into decimal. Don't tell your students how to do this, as there are plenty of good references to be found on the procedure. If a student cannot research and understand how to convert binary into decimal without your help, they aren't trying hard enough!

Question 11

Convert the following numbers from decimal (base-ten) to binary (base-two):

- $7_{10} =$
- $10_{10} =$
- $19_{10} =$
- $250_{10} =$
- $511_{10} =$
- $824_{10} =$
- $1044_{10} =$
- $9241_{10} =$

Describe a general, step-by-step procedure for converting decimal numbers into binary numbers.
file 01204

Answer 11

- $7_{10} = 111_2$
- $10_{10} = 1010_2$
- $19_{10} = 10011_2$
- $250_{10} = 11111010_2$
- $511_{10} = 11111111_2$
- $824_{10} = 1100111000_2$
- $1044_{10} = 10000010100_2$
- $9241_{10} = 10010000011001_2$

Notes 11

The most important part of this question, of course, is the *method* of translating decimal into binary. Don't tell your students how to do this, as there are plenty of good references to be found on the procedure. If a student cannot research and understand how to convert decimal into binary without your help, they aren't trying hard enough!

Question 12

A numeration system often used as a "shorthand" way of writing large binary numbers is the *octal*, or base-eight, system.

Based on what you know of place-weighted numeration systems, describe how many valid ciphers exist in the octal system, and the respective "weights" of each place in an octal number.

Also, perform the following conversions:

- 35_8 into decimal:
- 16_{10} into octal:
- 110010_2 into octal:
- 51_8 into binary:

file 01286

Answer 12

There are only eight valid ciphers in the octal system (0, 1, 2, 3, 4, 5, 6, and 7), with each successive place carrying eight times the "weight" of the place before it.

- 35_8 into decimal: 29_{10}
- 16_{10} into octal: 20_8
- 110010_2 into octal: 62_8
- 51_8 into binary: 101001_2

Follow-up question: why is octal considered a "shorthand" notation for binary numbers?

Notes 12

There are many references from which students may learn to perform these conversions. Your assistance should be minimal, as these procedures are simple to comprehend and easy to find.

Question 13

A numeration system often used as a "shorthand" way of writing large binary numbers is the *hexadecimal*, or base-sixteen, system.

Based on what you know of place-weighted numeration systems, describe how many valid ciphers exist in the hexadecimal system, and the respective "weights" of each place in a hexadecimal number.

Also, perform the following conversions:

- 35_{16} into decimal:
- 34_{10} into hexadecimal:
- 11100010_2 into hexadecimal:
- 93_{16} into binary:

file 01206

Answer 13

There are sixteen valid ciphers in the hexadecimal system (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, and F), with each successive place carrying sixteen times the "weight" of the place before it.

- 35_{16} into decimal: 53_{10}
- 34_{10} into hexadecimal: 22_{16}
- 11100010_2 into hexadecimal: $E2_{16}$
- 93_{16} into binary: 10010011_2

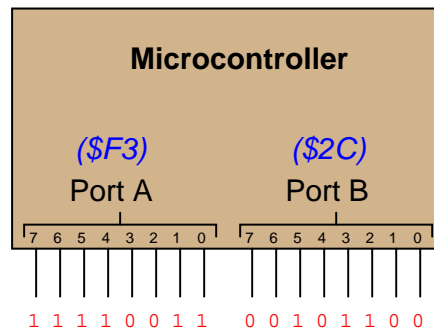
Follow-up question: why is hexadecimal considered a "shorthand" notation for binary numbers?

Notes 13

There are many references from which students may learn to perform these conversions. Your assistance should be minimal, as these procedures are simple to comprehend and easy to find.

Question 14

Digital computers communicate with external devices through *ports*: sets of terminals usually arranged in groups of 4, 8, 16, or more (4 bits = 1 *nybble*, 8 bits = 1 *byte*, 16 bits = 2 bytes). These terminals may be set to high or low logic states by writing a program for the computer that sends a numerical value to the port. For example, here is an illustration of a microcontroller being instructed to send the hexadecimal number F3 to port A and 2C to port B:



Suppose we wished to use the upper four bits of port A (pins 7, 6, 5, and 4) to drive the coils of a stepper motor in this eight-step sequence:

- Step 1: 0001
- Step 2: 0011
- Step 3: 0010
- Step 4: 0110
- Step 5: 0100
- Step 6: 1100
- Step 7: 1000
- Step 8: 1001

As each pin goes high, it drives a power MOSFET on, which sends current through that respective coil of the stepper motor. By following a "shift" sequence as shown, the motor will rotate a small amount for each cycle.

Write the necessary sequence of numbers to be sent to port A to generate this specific order of bit shifts, in hexadecimal. Leave the lower four bit of port A all in the low logic state.

file 02895

Answer 14

- Step 1: 10₁₆
- Step 2: 30₁₆
- Step 3: 20₁₆
- Step 4: 60₁₆
- Step 5: 40₁₆
- Step 6: C0₁₆
- Step 7: 80₁₆
- Step 8: 90₁₆

Follow-up question: write the same sequence in decimal rather than hexadecimal:

Step 1:
Step 2:
Step 3:
Step 4:
Step 5:
Step 6:
Step 7:
Step 8:

Notes 14

Although the root of this question is nothing more than binary-to-hexadecimal conversion, it also introduces students to the concept of controlling bit states in microcomputer ports by writing hex values. As such, this question is *very* practical!

In case students ask, let them know that a dollar sign prefix is sometimes used to denote a hexadecimal number. Other times, the prefix 0x is used (e.g., \$F3 and 0xF3 mean the same thing).

Question 15

Complete this table, performing all necessary conversions between numeration systems:

Binary	Octal	Decimal	Hexadecimal
10010			
		92	
			1A
	67		
1100101			
			122
		1000	
	336		
1011010110			

file 01207

Answer 15

Binary	Octal	Decimal	Hexadecimal
10010	22	18	12
1011100	134	92	5C
11010	32	26	1A
110111	67	55	37
1100101	145	101	65
100100010	442	290	122
1111101000	1750	1000	3E8
11011110	336	222	DE
1011010110	1326	726	2D6

Lots of conversions to do here! I particularly like the "table" format shown here for practicing numeration system conversions, because it compresses a lot of practice into a small space on paper, and also because it allows students to use different methods of conversion. For example, in converting a decimal number to the other forms, a student might choose to convert first to binary, then from binary to octal and hex. Or, alternatively, a student may choose to convert from decimal into hex, then from hex into binary, then from binary to octal, the last two conversions being especially easy.

Question 16

When representing non-whole numbers, we extend the "places" of our decimal numeration system past the right of the decimal point, like this:

<i>Decimal place-weights</i>								
$\frac{2}{10^3}$	$\frac{5}{10^2}$	$\frac{9}{10^1}$	$\frac{6}{10^0}$	•	$\frac{3}{10^{-1}}$	$\frac{8}{10^{-2}}$	$\frac{0}{10^{-3}}$	$\frac{4}{10^{-4}}$

$$2 \times 10^3 = 2000$$

$$3 \times 10^{-1} = \frac{3}{10}$$

$$5 \times 10^2 = 500$$

$$8 \times 10^{-2} = \frac{8}{100}$$

$$9 \times 10^1 = 90$$

$$0 \times 10^{-3} = \frac{0}{1000}$$

$$6 \times 10^0 = 6$$

$$4 \times 10^{-4} = \frac{4}{10000}$$

How do you suppose we represent non-whole numbers in a numeration system with a base (or "radix") other than ten? In the following examples, write the place-weight values underneath each place, and then determine the decimal equivalent of each example number:

<i>Binary place-weights</i>								
$\frac{1}{2^3}$	$\frac{0}{2^2}$	$\frac{0}{2^1}$	$\frac{1}{2^0}$	•	$\frac{1}{2^{-1}}$	$\frac{0}{2^{-2}}$	$\frac{1}{2^{-3}}$	$\frac{1}{2^{-4}}$

<i>Octal place-weights</i>								
$\frac{4}{8^3}$	$\frac{0}{8^2}$	$\frac{2}{8^1}$	$\frac{7}{8^0}$	•	$\frac{3}{8^{-1}}$	$\frac{6}{8^{-2}}$	$\frac{1}{8^{-3}}$	$\frac{2}{8^{-4}}$

<i>Hexadecimal place-weights</i>								
<u>C</u>	<u>1</u>	<u>A</u>	<u>6</u>	•	<u>3</u>	<u>2</u>	<u>B</u>	<u>9</u>

<i>Binary place-weights</i>							
$1001.1011_2 = 9.6875_{10}$							
$\frac{1}{2^3}$	$\frac{0}{2^2}$	$\frac{0}{2^1}$	$\frac{1}{2^0}$	$\bullet \frac{1}{2^{-1}}$	$\frac{0}{2^{-2}}$	$\frac{1}{2^{-3}}$	$\frac{1}{2^{-4}}$

<i>Octal place-weights</i>							
$4027.3612_8 = 2071.471191406_{10}$							
$\frac{4}{8^3}$	$\frac{0}{8^2}$	$\frac{2}{8^1}$	$\frac{7}{8^0}$	$\bullet \frac{3}{8^{-1}}$	$\frac{6}{8^{-2}}$	$\frac{1}{8^{-3}}$	$\frac{2}{8^{-4}}$

<i>Hexadecimal place-weights</i>							
$C1A6.32B9_{16} = 49574.198135376_{10}$							
$\frac{C}{16^3}$	$\frac{1}{16^2}$	$\frac{A}{16^1}$	$\frac{6}{16^0}$	$\bullet \frac{3}{16^{-1}}$	$\frac{2}{16^{-2}}$	$\frac{B}{16^{-3}}$	$\frac{9}{16^{-4}}$

Notes 16

Many students will not realize initially that it is possible to represent non-whole numbers in binary, octal, or hexadecimal. Really, though, the concept is identical to the representation of non-whole numbers in decimal form. The ability of your students to grasp non-whole numbers in these other numeration systems indicates their grasp of place-weighted systems in general. If a student truly comprehends how place-weighting works, they will have no trouble understanding digits to the left *or* the right of the radix point in any numeration system. If a student does not understand how digits to the right of the decimal point are interpreted in other numeration systems, then they need to spend more time reviewing what decimal numbers mean.

It's not that the concept is so hard to understand, so much as it is our familiarity with the decimal (base-ten) numeration system. We grow so accustomed to one way of representing numbers that we don't realize what the symbols actually *mean*, or that there may be alternative methods of representing quantities.

Question 17

Convert the following numbers (all between the values of 0 and 1) into decimal form:

- $0.001_2 =$
- $0.101_2 =$
- $0.10111_2 =$
- $0.005_8 =$
- $0.347_8 =$
- $0.34071_8 =$
- $0.00C_{16} =$
- $0.A2F_{16} =$
- $0.A2F09_{16} =$

file 01209

Answer 17

- $0.001_2 = 0.125_{10}$
- $0.101_2 = 0.625_{10}$
- $0.10111_2 = 0.71875_{10}$
- $0.005_8 = 0.009765625_{10}$
- $0.347_8 = 0.451171875_{10}$
- $0.34071_8 = 0.439239502_{10}$
- $0.00C_{16} = 0.002929688_{10}$
- $0.A2F_{16} = 0.636474609_{10}$
- $0.A2F09_{16} = 0.636483192_{10}$

Notes 17

Ask your students to explain the method of conversion used in each case. It is rather simple, but important to understand nonetheless. Ask your students how this method compares with the conversion of whole-number values into decimal form.

Question 18

Complete this table, performing all necessary conversions between numeration systems. Truncate all answers to three characters past the point:

Binary	Octal	Decimal	Hexadecimal
101.011			
		25.2	
			4.B
	72.52		
1011.101			
			AC.11
		934.79	
	641.7		
101100.1			

file 01210

Answer 18

Binary	Octal	Decimal	Hexadecimal
101.011	5.3	5.375	5.6
11001.001	31.146	25.2	19.333
100.101	4.54	4.687	4.B
111010.101	72.52	58.656	3A.A8
1011.101	13.5	11.625	B.A
10101100.000	254.042	172.066	AC.11
1110100110.110	1646.624	934.79	3A6.CA3
110100001.111	641.7	417.875	1A1.E
101100.1	54.4	44.5	2C.8

Lots of conversions to do here! I particularly like the "table" format shown here for practicing numeration system conversions, because it compresses a lot of practice into a small space on paper, and also because it allows students to use different methods of conversion. For example, in converting a decimal number to the other forms, a student might choose to convert first to binary, then from binary to octal and hex. Or, alternatively, a student may choose to convert from decimal into hex, then from hex into binary, then from binary to octal, the last two conversions being especially easy.

Question 19

In digital computer systems, binary numbers are often represented by a fixed number of bits, such as 8, or 16, or 32. Such bit groupings are often given special names, because they are so common in digital systems:

- byte
- nybble
- word

How many binary bits is represented by each of the above terms?

And, for those looking for more challenge, try defining these terms:

- nickle
- deckle
- chawmp
- playte
- dynner

[file 01223](#)

Answer 19

- byte = 8 bits
- nybble = 4 bits
- word = *depends on the system*

The term "word," is often used to represent 16 bits, but it really depends on the particular system being spoken of. A binary "word" is more accurately defined as the default width of a binary bit grouping in a digital system.

Follow-up question: what binary grouping corresponds to a single hexadecimal character?

Notes 19

Definitions taken from *The New Hacker's Dictionary*, available on internet terminals everywhere.