```
The myth of Niles and Pepper
 Niles was an unusual boy with great hair, living
 in a time before number systems. Quantities
 were familiar, but symbolic representations of
 them were not. Niles lived in a grove of trees. A
  grove on the other side of the hill was home to
  his friend Pepper, a sassy, no-nonsense girl.
 One day Niles and Pepper were walking
  together and, as children do, began arguing
  about whose grove had more trees. If the
 argument had been about who had more
 skipping stones, they could have simply
  matched up stone-for-stone to discover who
 had more. But this was impractical with the
 trees. Niles had an insight:
         We can represent each tree by a
         drawing and match these to
         determine who has more.
 It went something like this.
   Niles: *********
   Pepper: ***********
  Not to be discouraged, Niles proposed they
 compare, instead, the number of trees on each's
 entire side of the hill. However, there were
  many more trees, so Pepper suggested they
  simply draw the symbol \top to represent each
 tree, to save time. The results were no
  more-satisfying to Niles.
   Niles pushed on: let's include the neighboring
 hill on each side. With so many more trees,
 Pepper suggested a shorthand notation.
         We can compactly represent the
         number of trees with two symbols \bigcirc
         and | used in combination.
 She explained it to Niles by counting up:
      • = ()

    T = |

      • TT = |

    TTT = ||

      • TTTT = | \( \)
      • TTTTT = | O |

    TTTTTT = ||○

      • TTTTTTT = \parallel \mid.
 That is, each position could take on two
 symbols, but the position of each symbol
  denoted its "weight." The far-right symbol
  represented either a lack ○ or presence | of a
  single tree. The next symbol to the left was the
  "overflow" from the first symbol, and therefore
  represented the number of pairs of trees. The
 next symbol to the left was the next overflow,
 representing pairs of pairs of trees.
 What fun! They started counting and Pepper
 immediately recognized a process
 improvement:
         If we use more symbols, we can
         represent numbers even
         more-compactly.
 Pepper suggested a symbol for each digit of
 their hands:
                     0, 1, 2, 3, 4, 5, 6, 7, 8, 9.
 Now they could count on their fingers:
      • = 0

    T = 1

    TT = 2

    TTT = 3

    TTTT = 4

    TTTTT = 5

      • TTTTTT = 6

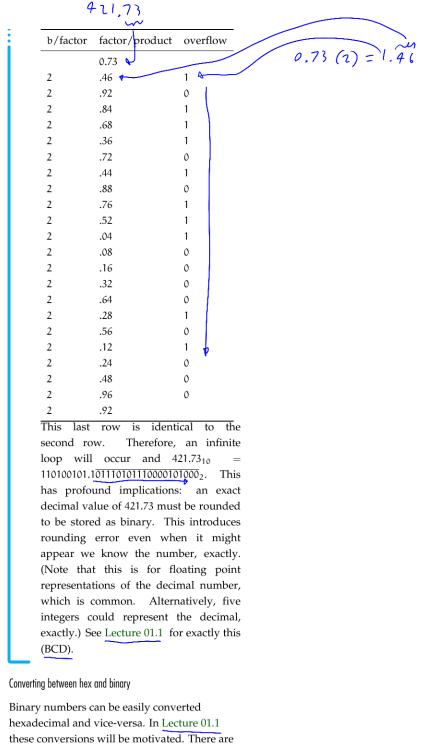
    TTTTTTTT = 8

      • TTTTTTTTT = 9
      • \top = 10.
 That is, the second symbol now represented a
 group of ten of the group to the right: the
 rightmost, the number of trees; to its left, the
 number of tens of trees; to its left, the number of
 tens of tens (hundreds); to its left, the number of
 tens of hundreds (thousands); etc.
 Pepper still had more trees.
 Positional numeral systems
 Niles and Pepper, when they represented a tree
 by \top, created a very simple numeral system: a
  way of representing quantities with symbols
 called numerals. Once they recognized the
 value of including multiple symbols (○ and |, at
 first) and endowed the position of each numeral
  with significance, the system became a
  positional numeral system. The number of
  numerals used is called the system's base: two
                                                                         binary
  for the system with \bigcirc and | and ten for that with
 0–9. Base-2 systems are called binary, and
  typically use the symbols <u>0</u> and <u>1</u> instead of ○
  and |. Base-10 systems are those with ten
  numerals, the most common of which is called
 the Hindu-Arabic numeral system and uses the
 Arabic numerals 0-9.
 Let's consider the meaning of the Arabic
 number 937. It means 9 hundreds, 3 tens, and 7
 ones. A corresponding arithmetic
 representation is
                 9 \times 10^2 + 3 \times 10^1 + 7 \times 10^0.
 Similarly, the binary number 1011 has the
 meaning 1 pair of pairs of pairs of ones, 0 pair of
 pairs of ones, 1 pair of ones, 1 one. Let's convert
 this binary representation to base-10:
 can we tell in which numeral system 11 is
                                                                          11,0 7 11,2
 written? We cannot, so we must either rely on
 context, explicitly state, or add subscripts, as in
                          1011_2 = 11_{10}.
  As a convention, we restrict interpretations of
 numeral system-denoting subscripts to base-10.
 Now we introduce nuanced versions of the
 above numeral systems.
 Decimal numeral system
  Representing non-integer numbers is done with
                                                                             decimal point
 a radix point, often ".". The decimal numeral
 system is the Hindu-Arabic system extended to
  include non-integer numbers. Digits (decimal
 numerals) increasingly right of the radix point
 (called a decimal point in the decimal system)
  represent tenths, hundredths, thousands, etc.
 For instance, the decimal 2.73 would be
               2 \times 10^{0} + 7 \times 10^{-1} + 3 \times 10^{-2}.
                1 + 0,7 0.03 = 2,73
  Hexadecimal numeral system
 The hexadecimal numeral system extends the
 decimal system with an additional six numerals,
 borrowed from the beginning of the Latin
 alphabet, to have a total of sixteen numerals:
           0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.
                                                                             24=16
 As we will see, this base-16 system provides a
 convenient way to represent the contents of
 computer memory.
                                                                           10112 = 0xB
 It is conventional to begin hexadecimal numbers
 with the prefix "0x" as in 0x9A780D38.
                                                                           1001 10112 = 0x 9B
 Converting to and from decimal
 Converting from a base-b number x_b with digits
 x_n x_{n-1} \cdots x_0 \cdot x_{-1} x_{-2} \cdots x_{-m} to decimal is
 straightforward. Represent each numeral in
base-10, then use the formula
x_{10} = \sum_{i=-m}^{n} x_i b^i.
(3)
|x|^6 + 9 \times 16^{i} = |1 + 9 \times 16 = |1 + 177 = |155|
For instance, if x_2 = 1010.01,
x_{10} = 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 + 0 \times 2^{-1} + 1 \times 2^{-2}
|x|^7 + o \times 1^6 + o \times 1^5 + 1 \times 1^4 + 1 \times 1^3 + o \times 1^4 + 1 \times 1^3 + o \times 1^4 + 1 \times 1^
 Similarly, if x_{16} = B8.F,
          x_{10} = 11 \times 16^1 + 8 \times 16^0 + 15 \times 16^{-1}
               = 184.9375.
 Converting from decimal into a base-b numeral
  system can be accomplished by the following
  procedure.
      • For the integer part of the number,
          successively divide by the base b_{10},
          represented in base-10. The remainder x_b,
          represented in base-b, of each step is the
         base-b numeral in that position, from
         right-to-left.
       • For the decimal part of the number,
         successively multiplying by the base b_{10}.
         The overflow x_{-b} of 1_{10} and above, at each
         step, is the corresponding base-b numeral
         in that position, from left-to-right.
  Note that division and multiplication in the
  conversion process are the usual base-10
  versions. Technically, this process can be used
  for converting between other numeral systems,
 but it is not recommended due to our
 unfamiliarity with division and multiplication
 in these numeral systems.
 Example 00.4 -1
                                                                      re: decimal to binary and hex
       1. Convert 14<sub>10</sub> to binary.
       2. Convert 14<sub>10</sub> to hexadecimal.
      3. Convert 421.73<sub>10</sub> to binary.
      1. The following table shows the division.
            b/divisor dividend/quotient remainder
                             14
           Therefore, 14_{10} = 1110_2. = 3 + 4 + 1
       2. There is no need to divide because
                                                                                                 181,0 to hex
          14_{10} \leq 15_{10}, the number of hex numerals.
          Therefore, 14_{10} = E_{16}.
       3. For the integer part, the following table
                                                                                                                       dividend remainder
           shows the division.
            b/divisor dividend/quotient remainder
                                                                                                 16
                                                                                                                         11
                             421
                             210
                                                                                                 16
                             105
                             52
                                                                                                            0xB5 = B516
                             26
                             13
           Therefore, 421_{10} = 110100101_2. Now for
           the number right of the decimal point.
```

00.4

**Numeral systems** 

The following is a myth in the good sense of the



these conversions will be motivated. There are  $2^4 = 16$  unique four-numeral binary numbers and 16 hex characters (which is no coincidence). This allows us to write each grouping of four binary numerals, called a nibble, as a single hex character. It is often easiest to convert each nibble to base-10, then (trivially) to hex. For instance,  $1101_2$  is  $1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 13_{10}$ . The thirteenth hex numeral is D. Similarly, one can convert a hex numeral to a nibble by converting it first to decimal, then to

binary.

Signed binary numeral system

The signed binary numeral system, often called

The <u>signed binary</u> numeral system, often called the <u>two's complement</u> numeral system, is used to represent both positive and negative numbers in binary form. When encountering a signed binary number, first consider the leftmost

numeral: if it is 0, the number it represents is positive or zero and the usual binary-to-decimal conversion holds; if it is 1, the number it represents is negative and must undergo the two's complement operation before the usual binary-to-decimal conversion holds for its negation. The two's complement operation can be performed by flipping all the bits  $(0 \to 1)$  and  $(1 \to 0)$  and adding 1. For instance,

This result can be converted to decimal in the usual way:  $0010\,1010_2 = 42_{10}$ . Therefore  $1101\,0110_2 = -42_{10}$ . Of course, this means that an n-numeral in two's complement binary stores not (as would unsigned binary)

 $0, 1, \cdots 2^{n} - 1$ 

 $-2^{n-1}+1, -2^{n-1}+2, \cdots -1, 0, 1 \cdots 2^{n-1}.$ 

1101 0110  $\xrightarrow{\text{flip bits}}$  0010 1001  $\xrightarrow{\text{add one}}$  0010 1010.

In both unsigned and (two's complement) signed binary, however, a total of  $2^n$  numbers are represented by n binary numerals.

86 int int 0,..., 255

Unsignal int -127,..., 128

1101

8 4 2 1

BS16 = 101101017

B5/6

B16 = 10112

5/6 = 01012