The carry flag (CF) and overflow flag (OF) in the status flags portion of the CPU's rflags register allow us to detect when adding and subtracting binary numbers yields results that exceed the allocated number of bits for the data type.

Subtracting in the Decimal Number System

```
let borrow = 0
Repeat for i = 0, ,(N   1)
   If y_i   x_i
      Let difference_i = x_i  y_i
   Else
      Let j = i + 1
      While(x_j = 0) and (j < N)
        Add 1 to j
      If j = N
        Let borrow = 1
        Subtract 1 from j
        Add 10 to x_j
      While j > i
        Subtract 1 from x_j
        Subtract 1 from j
        Add 10 to x_j
          Let difference_i = x_i  y_i
```

YOUR TURN

- How many bits are required to store a single decimal digit? Invent a code for storing eight decimal digits in 32 bits. Using this code, does binary addition produce the correct results? You'll see such a code later in the book and some reasons for its usefulness.

- Develop an algorithm for adding fixed-width integers in the binary number system.

- Develop an algorithm for adding fixed-width integers in the hexadecimal number system.

- Develop an algorithm for subtracting fixed-width integers in the binary number system.

- Develop an algorithm for subtracting fixed-width integers in the hexadecimal number system.

Two's Complement
In mathematics, the complement of a quantity is the amount that must be added to make it "whole." When applying this concept to numbers, the definition of whole depends on the radix (or base) you're working in and the number of digits that you allow to represent the numbers. If x is an n-digit number in radix r, its radix comple-

ment, ¬x, is defined such that $x + \neg x = radix^n$, where $radix^n$ is 1 followed by n 0s. For example, if we're working with two-digit decimal numbers, then the radix complement of 37 is 63, because $37 + 63 = 102 = 100$. Another way of saying this is that adding a number to its radix complement results in 0 with a carry beyond the n digits.

Diminished radix complement
Another useful concept is the diminished radix complement, which is defined such that $x + diminished\_radix\_complement = radix^n - 1$. For example, the diminished radix complement of 37 is 62, because $37 + 62 = 10^2 - 1 = 99$. If you add a number to its diminished radix complement, the result is n of the largest digits in the radix—two 9s in this example of two digits in radix 10.

One hexadecimal digit	Four binary digits (bits)	Signed decimal
8	1000	8
9	1001	7
a	1010	6
b	1011	5
c	1100	4
d	1101	3
e	1110	2
f	1111	1
0	0000	0
1	0001	+1
2	0010	+2
3	0011	+3
4	0100	+4
5	0101	+5
6	0110	+6
7	0111	+7

- The high-order bit of each positive number is 0, and the high-order bit of each negative number is 1.

- Although changing the sign of (negating) a number is more complicated than simply changing the high-order bit, it is common to call the high-order bit the sign bit.

- The notation allows for one more negative number than positive numbers.

- The range of integers, x, that can be represented in this notation (with four bits) is
$$-8_10 \leq x \leq +7_10$$
or
$$-2^{(4-1)} \leq x \leq +(2^{(4-1)}-1)$$
The last observation can be generalized for n bits to the following:
$$-2^{(n-1)} \leq x \leq +(2^{(n-1)}-1)$$

When using two's complement notation, the negative of any n-bit integer, x, is defined as

$$x + (-x) = 2^n$$