**Exercise 2-1.** Write a program to determine the ranges of char, short, int, and long variables, both signed and unsigned, by printing appropriate values from standard headers and by direct computation. Harder if you compute them: determine the ranges of the various floating-point types.

**Exercise 2-2.** Write a loop equivalent to the for loop above without using && or ||.

**Exercise 2-3.** Write a function htoi(s), which converts a string of hexadecimal digits (including an optional 0x or 0X) into its equivalent integer value. The allowable digits are 0 through 9, a through f, and A through F.

**Exercise 2-4.** Write an alternative version of squeeze(s1,s2) that deletes each character in s1 that matches any character in the string s2.

**Exercise 2-5.** Write the function any(s1,s2), which returns the first location in a string s1 where any character from the string s2 occurs, or -1 if s1 contains no characters from s2. (The standard library function strpbrk does the same job but returns a pointer to the location.)

**Exercise 2-6.** Write a function setbits(x,p,n,y) that returns x with the n bits that begin at position p set to the rightmost n bits of y, leaving the other bits unchanged.

**Exercise 2-7.** Write a function invert(x,p,n) that returns x with the n bits that begin at position p inverted (i.e., 1 changed into 0 and vice versa), leaving the others unchanged.

**Exercise 2-8.** Write a function rightrot(x,n) that returns the value of the integer x rotated to the right by n positions.

**Exercise 2-9.** In a two's complement number system, x &= (x-1) deletes the rightmost 1-bit in x. Explain why. Use this observation to write a faster version of bitcount.

**Exercise 2-10.** Rewrite the function lower, which converts upper case letters to lower case, with a conditional expression instead of if-else.