declared within a block of code (such as a function) or globally visible by all blocks.

- **Operators in C.** C's operators can be categorized by the function they perform: assignment, arithmetic, bitwise manipulations, logical and relational tests. We can form expressions using variables and operators such that the expressions get evaluated according to precedence and associativity rules. Expressions are grouped into statements, which express the work the program is to perform.

- **Translating C Variables and Operators into LC-3 Code.** Using a symbol table to keep track of variable declarations, a compiler will allocate local variables for a function within an activation record for the function. The activation record for the function is pushed onto the run-time stack whenever the function is executed. Global variables in a program are allocated in the global data section.

### Exercises

12.1 Generate the compiler's symbol table for the following code. Assume all variables occupy one location in memory.

```c
{   double ff;
    char cc;
    int ii;
    char dd;
}
```

12.2 The following variable declaration appears in a program:

```c
int r;
```

a. If `r` is a local variable, to what value will it be initialized?
b. If `r` is a global variable, to what value will it be initialized?

12.3 What are the ranges for the following two variables if they are stored as 32-bit quantities?

```c
int plusOrMinus;
unsigned int positive;
```

12.4 Evaluate the following floating point literals. Write their values in standard decimal notation.

a. 1.11 E -11
b. -0.00021 E 4
c. 101.101 E 0
12.5 Write the LC-3 code that would result if the following local variable declarations were compiled using the LC-3 C compiler:

```c
char c = 'a';
int x = 3;
int y;
int z = 10;
```

12.6 For the following code, state the values that are printed out by each `printf` statement. The statements are executed in the order A, B, C, D.

```c
int t; /* This variable is global */
{
  int t = 2;
  printf("%d\n", t); /* A */
  {
    printf("%d\n", t); /* B */
    t = 3;
  }
  printf("%d\n", t); /* C */
}
{
  printf("%d\n", t); /* D */
}
```

12.7 Given that `a` and `b` are both integers where `a` and `b` have been assigned the values 6 and 9, respectively, what is the value of each of the following expressions? Also, if the value of `a` or `b` changes, give their new value.

- `a | b`
- `a || b`
- `a & b`
- `a && b`
- `!(a + b)`
- `a % b`
- `b / a`
- `a = b`
- `a = b = 5`
- `++a + b--`
- `a = (++b < 3) ? a : b`
- `a <<= b`

12.8 For the following questions, write a C expression to perform the following relational test on the character variable `letter`.

- Test if `letter` is any alphabetic character or a number.
- Test if `letter` is any character except an alphabetic character or a number.
12.9  

a. What does the following statement accomplish? The variable 
   letter is a character variable.
   
   \[ \text{letter} = ((\text{letter} >= 'a' && \text{letter} <= 'z') ? '!' : \text{letter}); \]

b. Modify the statement in (a) so that it converts lowercase to 
   uppercase.

12.10 Write a program that reads an integer from the keyboard and displays a 
1 if it is divisible by 3 or a 0 otherwise.

12.11 Explain the differences between the following C statements:

a. \( j = i++; \)

b. \( j = ++i; \)

c. \( j = i + 1; \)

d. \( i += 1; \)

e. \( j = i += 1; \)

f. Which statements modify the value of \( i \)? Which ones modify the 
   value of \( j \)? If \( i = 1 \) and \( j = 0 \) initially, what will the values of 
   \( i \) and \( j \) be after each statement is run separately?

12.12 Say variables \( a \) and \( b \) are both declared locally as \text{long int}.

a. Translate the expression \( a + b \) into LC-3 code, assuming a 
   \text{long int} occupies two bytes. Assume \( a \) is allocated at offset 0 and 
   \( b \) is at offset \(-1\) in the activation record for their function.

b. Translate the same expression, assuming a \text{long int} occupies four 
   bytes, \( a \) is allocated offset 3, and \( b \) is at offset 5.

12.13 If initially, \( a = 1, b = 1, c = 3, \) and \( \text{result} = 999 \), what are the 
values of the variables after the following C statement is executed?

\[ \text{result} = b + 1 | c + a; \]

12.14 Recall the machine busy example from previous chapters. Say the 
integer variable \text{machineBusy} tracks the busyness of all 16 machines. 
Recall that a 0 in a particular bit position indicates the machine is busy 
and a 1 in that position indicates the machine is idle.

a. Write a C statement to make machine 5 busy.

b. Write a C statement to make machine 10 idle.

c. Write a C statement to make machine \( n \) busy. That is, the machine 
   that has become busy is an integer variable \( n \).

d. Write a C expression to check if machine 3 is idle. If it is idle, the 
   expression returns a 1. If it is busy, the expression returns a 0.

e. Write a C expression that evaluates to the number of idle machines. 
   For example, if the binary pattern in \text{machineBusy} were 
   \text{1011 0010 1110 1001}, then the expression will evaluate to 9.
12.15 What purpose does the semicolon serve in C?

12.16 Say we are designing a new computer programming language that includes the operators @, #, $ and \$. How would the expression
\[ w \@ x \# y \$ z \$ \] get evaluated under the following constraints?
   a. The precedence of @ is higher than # is higher than $ is higher than \$. Use parentheses to indicate the order.
   b. The precedence of # is higher than \$ is higher than @ is higher than $.
   c. Their precedence is all the same, but they associate left to right.
   d. Their precedence is all the same, but they associate right to left.

12.17 Notice that the C assignment operators have the lowest precedence. Say we have developed a new programming language called Q that works exactly like C, except that the assignment operator had the highest precedence.
   a. What is the result of the following Q statement? In other words, what would the value of x be after it executed?
      \[ x = x + 1; \]
   b. How would we change this Q statement so that it works the same way as it would in C?

12.18 Modify the example program in Chapter 11 (Figure 11.3) so that it prompts the user to type a character and then prints every character from that character down to the character in the order they appear in the ASCII table.

12.19 Write a C program to calculate the sales tax on a sales transaction. Prompt the user to enter the amount of the purchase and the tax rate. Output the amount of sales tax and the total amount (including tax) on the whole purchase.

12.20 Suppose a program contains the two integer variables x and y, which have values 3 and 4, respectively. Write C statements that will exchange the values in x and y so that after the statements are executed, x is equal to 4 and y is equal to 3.
   a. First, write this routine using a temporary variable for storage.
   b. Now rewrite this routine without using a temporary variable.