Exercises

Section 2.1

1. Create and run Kernighan and Ritchie's famous "hello, world" program:

```
#include <stdio.h>

int main(void)
{
  printf("hello, world\n");
}
```

Do you get a warning message from the compiler? If so, what's needed to make it go away?

Section 2.2

W 2. Consider the following program:

```
#include <stdio.h>

int main(void)
{
  printf("Parkinson's Law:\nWork expands so as to ");
  printf("fill the time\n");
  printf("available for its completion.\n");
  return 0;
}
```

(a) Identify the directives and statements in this program.
(b) What output does the program produce?

Section 2.4

W 3. Condense the `dweight.c` program by (1) replacing the assignments to `height`, `length`, and `width` with initializers and (2) removing the `weight` variable, instead calculating `(volume + 165) / 166` within the last `printf`.

W 4. Write a program that declares several `int` and `float` variables—without initializing them—and then prints their values. Is there any pattern to the values? (Usually there isn't.)

Section 2.7

W 5. Which of the following are not legal C identifiers?

(a) `100_bottles`
(b) `_100_bottles`
(c) `one__hundred__bottles`
(d) `bottles_by_the_hundred_`

6. Why is it not a good idea for an identifier to contain more than one adjacent underscore (as in `current___balance`, for example)?

7. Which of the following are keywords in C?

(a) `for`
(b) `If`
(c) `main`
(d) `printf`
(e) `while`

W Answer available on the Web at *knking.com/books/c2*.

Section 2.8

W 8. How many tokens are there in the following statement?

```
answer=(3*q-p*p)/3;
```

9. Insert spaces between the tokens in Exercise 8 to make the statement easier to read.

10. In the `dweight.c` program (Section 2.4), which spaces are essential?

Programming Projects

1. Write a program that uses `printf` to display the following picture on the screen:

```
    *
   *
  *
*   *
 * *
  *
```

2. Write a program that computes the volume of a sphere with a 10-meter radius, using the formula $v = 4/3\pi r^3$. Write the fraction 4/3 as `4.0f/3.0f`. (Try writing it as `4/3`. What happens?) *Hint:* C doesn't have an exponentiation operator, so you'll need to multiply r by itself twice to compute r^3.

3. Modify the program of Programming Project 2 so that it prompts the user to enter the radius of the sphere.

W 4. Write a program that asks the user to enter a dollars-and-cents amount, then displays the amount with 5% tax added:

```
Enter an amount: 100.00
With tax added: $105.00
```

5. Write a program that asks the user to enter a value for x and then displays the value of the following polynomial:

$3x^5 + 2x^4 - 5x^3 - x^2 + 7x - 6$

Hint: C doesn't have an exponentiation operator, so you'll need to multiply x by itself repeatedly in order to compute the powers of x. (For example, `x * x * x` is `x` cubed.)

6. Modify the program of Programming Project 5 so that the polynomial is evaluated using the following formula:

$((((3x + 2)x - 5)x - 1)x + 7)x - 6$

Note that the modified program performs fewer multiplications. This technique for evaluating polynomials is known as ***Horner's Rule***.

7. Write a program that asks the user to enter a U.S. dollar amount and then shows how to pay that amount using the smallest number of $20, $10, $5, and $1 bills:

```
Enter a dollar amount: 93

$20 bills: 4
$10 bills: 1
 $5 bills: 0
 $1 bills: 3
```