Exercises

Section 6.1

1. What output does the following program fragment produce?

```
i = 1;
while (i <= 128) {
  printf("%d ", i);
  i *= 2;
}
```

Section 6.2

2. What output does the following program fragment produce?

```
i = 9384;
do {
  printf("%d ", i);
  i /= 10;
} while (i > 0);
```

Section 6.3

*3. What output does the following `for` statement produce?

```
for (i = 5, j = i - 1; i > 0, j > 0; --i, j = i - 1)
  printf("%d ", i);
```

(W) 4. Which one of the following statements is not equivalent to the other two (assuming that the loop bodies are the same)?

(a) `for (i = 0; i < 10; i++) …`
(b) `for (i = 0; i < 10; ++i) …`
(c) `for (i = 0; i++ < 10; ) …`

5. Which one of the following statements is not equivalent to the other two (assuming that the loop bodies are the same)?

(a) `while (i < 10) {…}`
(b) `for (; i < 10;) {…}`
(c) `do {…} while (i < 10);`

6. Translate the program fragment of Exercise 1 into a single `for` statement.

7. Translate the program fragment of Exercise 2 into a single `for` statement.

*8. What output does the following `for` statement produce?

```
for (i = 10; i >= 1; i /= 2)
  printf("%d ", i++);
```

9. Translate the `for` statement of Exercise 8 into an equivalent `while` statement. You will need one statement in addition to the `while` loop itself.

Section 6.4

(W) 10. Show how to replace a `continue` statement by an equivalent `goto` statement.

11. What output does the following program fragment produce?

```
sum = 0;
for (i = 0; i < 10; i++) {
  if (i % 2)
    continue;
  sum += i;
}
printf("%d\n", sum);
```

W 12. The following "prime-testing" loop appeared in Section 6.4 as an example:

```
for (d = 2; d < n; d++)
  if (n % d == 0)
    break;
```

This loop isn't very efficient. It's not necessary to divide `n` by all numbers between 2 and `n` – 1 to determine whether it's prime. In fact, we need only check divisors up to the square root of `n`. Modify the loop to take advantage of this fact. *Hint:* Don't try to compute the square root of `n`; instead, compare `d * d` with `n`.

Section 6.5

*13. Rewrite the following loop so that its body is empty:

```
for (n = 0; m > 0; n++)
  m /= 2;
```

W*14. Find the error in the following program fragment and fix it.

```
if (n % 2 == 0);
  printf("n is even\n");
```

Programming Projects

1. Write a program that finds the largest in a series of numbers entered by the user. The program must prompt the user to enter numbers one by one. When the user enters 0 or a negative number, the program must display the largest nonnegative number entered:

```
Enter a number: 60
Enter a number: 38.3
Enter a number: 4.89
Enter a number: 100.62
Enter a number: 75.2295
Enter a number: 0

The largest number entered was 100.62
```

Notice that the numbers aren't necessarily integers.

W 2. Write a program that asks the user to enter two integers, then calculates and displays their greatest common divisor (GCD):

```
Enter two integers: 12 28
Greatest common divisor: 4
```

Hint: The classic algorithm for computing the GCD, known as Euclid's algorithm, goes as follows: Let `m` and `n` be variables containing the two numbers. If `n` is 0, then stop: `m` contains the GCD. Otherwise, compute the remainder when `m` is divided by `n`. Copy `n` into `m` and copy the remainder into `n`. Then repeat the process, starting with testing whether `n` is 0.