Section 5.2

Self Check

2. What values are displayed if the call to WriteLn comes at the end of the loop instead of at the beginning?

Programming

1. When Robin’s new baby was born, she opened a savings account with $1000.00. On each birthday, starting with the first, the bank added an additional 4.5% of the balance and Robin added another $500.00 to the account. Write a loop that will determine how much money was in the account on the child’s 18th birthday.

Section 5.3

Self Check

2. When would the output of the following segment be erroneous? How could it be fixed?

```pascal
Total := 0;
Write ('Enter number of items> '); ReadLn (Num);
Count := 0;
while Count < Num do
begin
    Write ('Enter a value> '); ReadLn (Value);
    Last := Value
end; {while}
WriteLn ('The last value entered was ', Last)
```

Programming

1. There are 9870 people in a town whose population increases by 10% each year. Write a loop that determines how many years (CountYears) it would take for the population to exceed 30,000.
Section 5.4

Self Check

2. Why would it be incorrect to move the assignment statement in the sentinel-controlled loop of Fig. 5.8 to the end of the loop body?

Programming

2. Write a program segment that allows the user to enter values and outputs the number of positive values entered and the number of negative values entered. Use 0 as the sentinel value.

4. Write a loop that prints a table of angle measures along with their sine and cosine values. Assume that the initial and final angle measures (in degrees) are available in InitDeg and FinalDeg (type Real), respectively, and that the change in angle measure between table entries is given by StepDeg. Hint: Don’t forget to change degrees to radians.

5. Write a flag-controlled loop that continues to read pairs of integers until it reads a pair with the property that the first integer in the pair is evenly divisible by the second.

Section 5.5

Self Check

2. What is the minimum number of times the statement that comprises the for loop body can be executed? Give an example program fragment that will execute that minimum number of times.

Programming

2. Write a program fragment with a for statement that accumulates the total number of days for the years 1950 to the year 2000. Remember, any year divisible by 4 is a leap year and has 366 days.
Section 5.6

Self Check

2. What would the following repeat statement display? What is the difference between repeat ... until False and while False do?

```plaintext
repeat
  WriteLn ('False conditional example.');
until False
```

Programming

1. Write a program fragment that continues to read data values as long as they are not decreasing. The fragment should stop reading whenever a number smaller than the preceding one is entered. Write two versions: one with repeat and one with while.

Section 5.7

Self Check

2. Show the output printed by the following nested loops:

```plaintext
for I := 1 to 2 do
  begin
    WriteLn ('Outer' :5, I :5);
    for J := 1 to 3 do
      WriteLn ('Inner' :7, I :3, J :3)
    for K := 2 downto 1 do
      WriteLn ('Inner' :7, I :3, K :3)
  end {for I}
```
Programming

1. Write a program fragment that, given an input value $N$, displays $N$ rows of the form 1 2 ... $N$, 2 3 ... $N + 1$, and so on—a barber pole pattern of numbers. As an example, for an input value of 5, display:

   1 2 3 4 5
   2 3 4 5 6
   3 4 5 6 7
   4 5 6 7 8
   5 6 7 8 9

Section 5.8

Self Check

2. Why does the following code fragment fail? What are the loop boundaries for this code fragment?

   ```
   X := 10;
   repeat
     X := X - 1;
     WriteLn (X, Sqrt(X))
   until X < 0
   ```

Chapter 5

Programming projects

2. Modify Programming Project 1 to compute and display both the range of values in the data collection and the standard deviation of the data collection. To compute the standard deviation, accumulate the sum of the data ($\text{Sum}$) and the sum of the squares of the data values ($\text{SumSquares}$) in the main loop. After loop exit, use the formula:

   $\text{Standard deviation} = \sqrt{\frac{\text{SumSquares} - \text{Sum}^2}{N}}$
4. Write a program to generate a yearly calendar. The program should accept the year and the day of the week for January 1 of that year (1 = Sunday, 7 = Saturday). Remember, February has 29 days if the year is divisible by 4. The calendar should be printed in the following form (for each month):

```
January
1
2  3  4  5  6  7  8
9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29
30 31
```

5. a. Write a program to read in a collection of exam scores ranging in value from 1 to 100. Your program should count and print the number of outstanding scores (90–100), the number of satisfactory scores (60–89), and the number of unsatisfactory scores (1–59). It should also display the category of each score. Test your program on the following data:

```
63 75 72 72 78 67 80 63 75
90 89 43 59 99 82 12 100
```

b. Modify your program to display the average exam score (a real number) at the end of the run.