Chapter 10

Loops

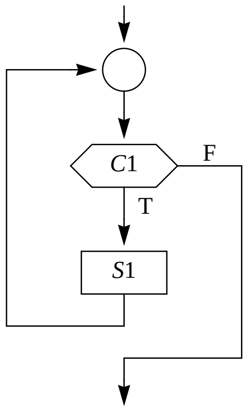
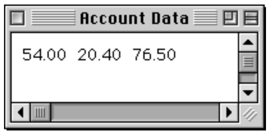


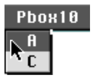
Figure 10.1
The flowchart for the WHILE statement.

```
WHILE C1 DO
   S1
END
```

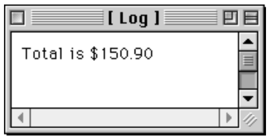


```
54.00  20.40  76.50
```

(a) The input window.



(b) The menu selection



```
Total is $150.90
```

(c) The output to the Log.

Figure 10.2
The input and output of the program in Listing 10.3.

```
MODULE Pbox10A;
    IMPORT TextModels, TextControllers, PboxMappers, PboxStrings, StdLog;

    PROCEDURE ComputeTotal*;
        VAR
            md: TextModels.Model;
            cn: TextControllers.Controller;
            sc: PboxMappers.Scanner;
            balance: REAL;
            sum: REAL;
            sumString: ARRAY 16 OF CHAR;
```

Figure 10.3
A program to find the total of all the data values in the focus window. It uses the eot technique.

```
    BEGIN
        cn := TextControllers.Focus();
        IF cn # NIL THEN
            md := cn.text;
            sc.ConnectTo(md);
            sum := 0.0;
            sc.ScanReal(balance);
            WHILE ~sc.eot DO
                sum := sum + balance;
                sc.ScanReal(balance)
            END;
            PboxStrings.RealToString(sum, 1, 2, sumString);
            StdLog.String("Total is $");
            StdLog.String(sumString); StdLog.Ln
        END
    END ComputeTotal;

END Pbox10A.
```

```
                                                              sum   balance   sc.eot   sumString
    cn := TextControllers.Focus();                        0
    IF cn # NIL THEN                                      0
        md := cn.text;                                    0
        sc.ConnectTo(md);                                 0
        sum := 0.0;                                       0
        sc.ScanReal(balance);                             0
        WHILE ~sc.eot DO                                  0
            sum := sum + balance;                         0
            sc.ScanReal(balance)                          0
        END;
        PboxStrings.RealToString(sum, 1, 2, sumString);   0
        StdLog.String("Total is $");                      0
        StdLog.String(sumString); StdLog.Ln               0
    END
```

```
Total                                                     0
```

```
                                                      sum    balance   sc.eot   sumString
▮  cn := TextControllers.Focus();                 1
   IF cn # NIL THEN                               0
      md := cn.text;                              0
      sc.ConnectTo(md);                           0
      sum := 0.0;                                 0
      sc.ScanReal(balance);                       0
      WHILE ~sc.eot DO                            0
         sum := sum + balance;                    0
         sc.ScanReal(balance)                     0
      END;
      PboxStrings.RealToString(sum, 1, 2, sumString);  0
      StdLog.String("Total is $");                0
      StdLog.String(sumString); StdLog.Ln         0
   END
```

```
Total                                             1
```

```
                                                      sum   balance   sc.eot   sumString
   cn := TextControllers.Focus();                1
   IF cn # NIL THEN                              1
      md := cn.text;                             0
      sc.ConnectTo(md);                          0
      sum := 0.0;                                0
      sc.ScanReal(balance);                      0
      WHILE ~sc.eot DO                           0
         sum := sum + balance;                   0
         sc.ScanReal(balance)                    0
      END;
      PboxStrings.RealToString(sum, 1, 2, sumString);   0
      StdLog.String("Total is $");               0
      StdLog.String(sumString); StdLog.Ln        0
   END
```

Total 2

```
                                                            sum    balance   sc.eot   sumString
   cn := TextControllers.Focus();                     1
   IF cn # NIL THEN                                   1
      md := cn.text;                                  1
      sc.ConnectTo(md);                               0
      sum := 0.0;                                     0
      sc.ScanReal(balance);                           0
      WHILE ~sc.eot DO                                0
         sum := sum + balance;                        0
         sc.ScanReal(balance)                         0
      END;
      PboxStrings.RealToString(sum, 1, 2, sumString); 0
      StdLog.String("Total is $");                    0
      StdLog.String(sumString); StdLog.Ln             0
   END
```

```
Total                                                 3
```

```
                                                          sum    balance   sc.eot   sumString
   cn := TextControllers.Focus();                      1
   IF cn # NIL THEN                                    1
      md := cn.text;                                   1
      sc.ConnectTo(md);                                1
      sum := 0.0;                                      0
      sc.ScanReal(balance);                            0
      WHILE ~sc.eot DO                                 0
         sum := sum + balance;                         0
         sc.ScanReal(balance)                          0
      END;
      PboxStrings.RealToString(sum, 1, 2, sumString);  0
      StdLog.String("Total is $");                     0
      StdLog.String(sumString); StdLog.Ln              0
   END

Total                                                  4
```

```
                                                     sum    balance   sc.eot   sumString
cn := TextControllers.Focus();                    1  0.0
IF cn # NIL THEN                                  1
    md := cn.text;                                1
    sc.ConnectTo(md);                             1
    sum := 0.0;                                   1
    sc.ScanReal(balance);                         0
    WHILE ~sc.eot DO                              0
        sum := sum + balance;                     0
        sc.ScanReal(balance)                      0
    END;
    PboxStrings.RealToString(sum, 1, 2, sumString); 0
    StdLog.String("Total is $");                  0
    StdLog.String(sumString); StdLog.Ln           0
END
```

Total　　5

```
                                                        sum    balance   sc.eot   sumString
   cn := TextControllers.Focus();                   1   0.0    54.0      false
   IF cn # NIL THEN                                 1
      md := cn.text;                                1
      sc.ConnectTo(md);                             1
      sum := 0.0;                                   1
▮     sc.ScanReal(balance);                         1
      WHILE ~sc.eot DO                              0
         sum := sum + balance;                      0
         sc.ScanReal(balance)                       0
      END;
      PboxStrings.RealToString(sum, 1, 2, sumString);  0
      StdLog.String("Total is $");                  0
      StdLog.String(sumString); StdLog.Ln           0
   END
```

```
Total                                               6
```

```
                                                              sum   balance   sc.eot   sumString
   cn := TextControllers.Focus();                        1    0.0   54.0      false
   IF cn # NIL THEN                                      1
      md := cn.text;                                     1
      sc.ConnectTo(md);                                  1
      sum := 0.0;                                        1
      sc.ScanReal(balance);                              1
▮     WHILE ~sc.eot DO                                   1
         sum := sum + balance;                           0
         sc.ScanReal(balance)                            0
      END;
      PboxStrings.RealToString(sum, 1, 2, sumString);    0
      StdLog.String("Total is $");                       0
      StdLog.String(sumString); StdLog.Ln                0
   END
________________________________________________________________________

Total                                                    7
```

```
                                                           sum     balance   sc.eot   sumString
   cn := TextControllers.Focus();                      1   54.0    54.0      false
   IF cn # NIL THEN                                    1
      md := cn.text;                                   1
      sc.ConnectTo(md);                                1
      sum := 0.0;                                      1
      sc.ScanReal(balance);                            1
      WHILE ~sc.eot DO                                 1
         sum := sum + balance;                         1
         sc.ScanReal(balance)                          0
      END;
      PboxStrings.RealToString(sum, 1, 2, sumString);  0
      StdLog.String("Total is $");                     0
      StdLog.String(sumString); StdLog.Ln              0
   END

Total                                                  8
```

```
                                                          sum     balance   sc.eot   sumString
   cn := TextControllers.Focus();                    1    54.0    20.4      false
   IF cn # NIL THEN                                  1
      md := cn.text;                                 1
      sc.ConnectTo(md);                              1
      sum := 0.0;                                    1
      sc.ScanReal(balance);                          1
      WHILE ~sc.eot DO                               1
         sum := sum + balance;                       1
▮        sc.ScanReal(balance)                        1
      END;
      PboxStrings.RealToString(sum, 1, 2, sumString); 0
      StdLog.String("Total is $");                   0
      StdLog.String(sumString); StdLog.Ln            0
   END
________________________________________________________________________

Total                                                9
```

```
                                                      sum     balance   sc.eot   sumString
  cn := TextControllers.Focus();                  1   54.0    20.4      false
  IF cn # NIL THEN                                1
      md := cn.text;                              1
      sc.ConnectTo(md);                           1
      sum := 0.0;                                 1
      sc.ScanReal(balance);                       1
      WHILE ~sc.eot DO                            2
          sum := sum + balance;                   1
          sc.ScanReal(balance)                    1
      END;
      PboxStrings.RealToString(sum, 1, 2, sumString); 0
      StdLog.String("Total is $");                0
      StdLog.String(sumString); StdLog.Ln         0
  END
```

Total 10

```
                                                    sum     balance   sc.eot   sumString
   cn := TextControllers.Focus();               1   74.4    20.4      false
   IF cn # NIL THEN                             1
      md := cn.text;                            1
      sc.ConnectTo(md);                         1
      sum := 0.0;                               1
      sc.ScanReal(balance);                     1
      WHILE ~sc.eot DO                          2
▮        sum := sum + balance;                  2
         sc.ScanReal(balance)                   1
      END;
      PboxStrings.RealToString(sum, 1, 2, sumString);   0
      StdLog.String("Total is $");              0
      StdLog.String(sumString); StdLog.Ln       0
   END
```

```
Total                                           11
```

```
                                                       sum     balance   sc.eot   sumString
  cn := TextControllers.Focus();                  1    74.4    76.5      false
  IF cn # NIL THEN                                1
     md := cn.text;                               1
     sc.ConnectTo(md);                            1
     sum := 0.0;                                  1
     sc.ScanReal(balance);                        1
     WHILE ~sc.eot DO                             2
        sum := sum + balance;                     2
        sc.ScanReal(balance)                      2
     END;
     PboxStrings.RealToString(sum, 1, 2, sumString);  0
     StdLog.String("Total is $");                 0
     StdLog.String(sumString); StdLog.Ln          0
  END
```

Total 12

```
                                                        sum     balance   sc.eot   sumString
   cn := TextControllers.Focus();                    1  74.4    76.5      false
   IF cn # NIL THEN                                  1
      md := cn.text;                                 1
      sc.ConnectTo(md);                              1
      sum := 0.0;                                    1
      sc.ScanReal(balance);                          1
▮     WHILE ~sc.eot DO                               3
         sum := sum + balance;                       2
         sc.ScanReal(balance)                        2
      END;
      PboxStrings.RealToString(sum, 1, 2, sumString); 0
      StdLog.String("Total is $");                   0
      StdLog.String(sumString); StdLog.Ln            0
   END
```

Total 13

```
                                                          sum     balance   sc.eot   sumString
   cn := TextControllers.Focus();                     1   150.9   76.5      false
   IF cn # NIL THEN                                   1
      md := cn.text;                                  1
      sc.ConnectTo(md);                               1
      sum := 0.0;                                     1
      sc.ScanReal(balance);                           1
      WHILE ~sc.eot DO                                3
▮        sum := sum + balance;                        3
         sc.ScanReal(balance)                         2
      END;
      PboxStrings.RealToString(sum, 1, 2, sumString); 0
      StdLog.String("Total is $");                    0
      StdLog.String(sumString); StdLog.Ln             0
   END
________________________________________________________

Total                                                 14
```

```
                                                          sum     balance   sc.eot   sumString
   cn := TextControllers.Focus();                    1    150.9   ?         true
   IF cn # NIL THEN                                  1
      md := cn.text;                                 1
      sc.ConnectTo(md);                              1
      sum := 0.0;                                    1
      sc.ScanReal(balance);                          1
      WHILE ~sc.eot DO                               3
         sum := sum + balance;                       3
         sc.ScanReal(balance)                        3
      END;
      PboxStrings.RealToString(sum, 1, 2, sumString);   0
      StdLog.String("Total is $");                   0
      StdLog.String(sumString); StdLog.Ln            0
   END
```

Total 15

```
                                                         sum     balance   sc.eot   sumString
  cn := TextControllers.Focus();                    1    150.9   ?         true
  IF cn # NIL THEN                                  1
     md := cn.text;                                 1
     sc.ConnectTo(md);                              1
     sum := 0.0;                                    1
     sc.ScanReal(balance);                          1
     WHILE ~sc.eot DO                               4
        sum := sum + balance;                       3
        sc.ScanReal(balance)                        3
     END;
     PboxStrings.RealToString(sum, 1, 2, sumString); 0
     StdLog.String("Total is $");                   0
     StdLog.String(sumString); StdLog.Ln            0
  END
```

Total 16

```
                                                           sum     balance   sc.eot   sumString
   cn := TextControllers.Focus();                      1   150.9   ?         true     “150.90”
   IF cn # NIL THEN                                    1
      md := cn.text;                                   1
      sc.ConnectTo(md);                                1
      sum := 0.0;                                      1
      sc.ScanReal(balance);                            1
      WHILE ~sc.eot DO                                 4
         sum := sum + balance;                         3
         sc.ScanReal(balance)                          3
      END;
      PboxStrings.RealToString(sum, 1, 2, sumString);  1
      StdLog.String("Total is $");                     0
      StdLog.String(sumString); StdLog.Ln              0
   END
```

Total 17

```
                                                         sum     balance   sc.eot   sumString
   cn := TextControllers.Focus();                     1  150.9   ?         true     “150.90”
   IF cn # NIL THEN                                   1
      md := cn.text;                                  1
      sc.ConnectTo(md);                               1
      sum := 0.0;                                     1
      sc.ScanReal(balance);                           1
      WHILE ~sc.eot DO                                4
         sum := sum + balance;                        3
         sc.ScanReal(balance)                         3
      END;
      PboxStrings.RealToString(sum, 1, 2, sumString); 1
      StdLog.String("Total is $");                    1
      StdLog.String(sumString); StdLog.Ln             0
   END
```

Total 18

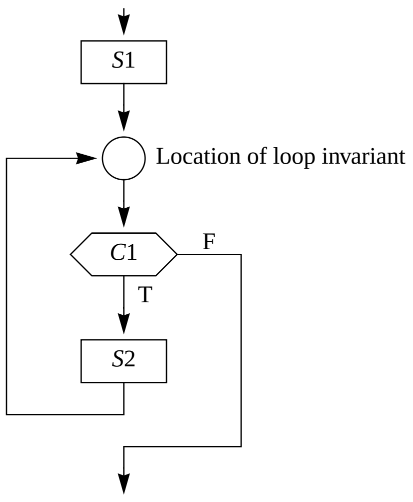
```
                                                     sum     balance   sc.eot   sumString
   cn := TextControllers.Focus();                 1  150.9   ?         true     “150.90”
   IF cn # NIL THEN                               1
      md := cn.text;                              1
      sc.ConnectTo(md);                           1
      sum := 0.0;                                 1
      sc.ScanReal(balance);                       1
      WHILE ~sc.eot DO                            4
         sum := sum + balance;                    3
         sc.ScanReal(balance)                     3
      END;
      PboxStrings.RealToString(sum, 1, 2, sumString);  1
      StdLog.String("Total is $");                1
      StdLog.String(sumString); StdLog.Ln         1
   END
```

Total 19

Statement	**No data values**	**Three data values**	***n* data values**
(1)	1	1	1
(2)	1	1	1
(3)	1	1	1
(4)	1	1	1
(5)	1	1	1
(6)	1	1	1
(7)	1	4	$n + 1$
(8)	0	3	n
(9)	0	3	n
(10)	1	1	1
(11)	1	1	1
(12)	1	1	1
Total:	10	19	$3n + 10$

Figure 10.4
Statement execution count for the procedure ComputeTotal in Figure 10.3.

Figure 10.5
The location of the loop invariant for a WHILE loop.



(a) Flowchart.

```
Statement 1
(* Location of loop invariant *)
(* Loop invariant is true. *)
WHILE Condition1 DO
    Statement2
END
(* Loop invariant is true and Condition1 is false. *)
```

(b) Source code.

```
BEGIN
    cn := TextControllers.Focus();
    IF cn # NIL THEN
        md := cn.text;
        sc.ConnectTo(md);
        sum := 0.0;
        sc.ScanReal(balance);
        (* Here is where the loop invariant is true *)
        WHILE ~sc.eot DO
            sum := sum + balance;
            sc.ScanReal(balance)
        END;
        PboxStrings.RealToString(sum, 1, 2, sumString);
        StdLog.String("Total is $");
        StdLog.String(sumString); StdLog.Ln
    END
END ComputeTotal;
```

The loop invariant for Figure 10.3

- sum is the total of all the values scanned, not including the current value scanned into balance.

To prove that a statement is a loop invariant, you must show two things:

Proving a loop invariant

- The statement is true initially because of the execution of *S*1.
- The statement is true at the end of each loop because of the execution of *S*2.

When a WHILE loop terminates

- The loop invariant is true.
- The loop condition is false.

PROCEDURE (VAR s: Scanner) **ScanReal** (OUT x: REAL), NEW
Pre
s is connected to a text model.   20
Characters scanned represent a real or integer value.   21
Post
~s.eot
    x gets the next real or integer value scanned.
s.eot
    x gets MAX(REAL)

PROCEDURE (VAR s: Scanner) **ScanInt** (OUT n: INTEGER), NEW
Pre
s is connected to a text model. 20
Characters scanned represent an integer value. 21
Post
~s.eot
    n gets the next integer value scanned.
s.eot
    n gets MAX(INTEGER)

```
sum := 0.0;
numAccts := 0;
sc.ScanReal(balance);
WHILE ~sc.eot DO
    sum := sum + balance;
    INC(numAccts);
    sc.ScanReal(balance)
END;
IF numAccts > 0 THEN
    Output sum / numAccts
ELSE
    Output a no accounts message
END
```

Figure 10.6
An algorithm to find the average of all the data values in the focus window.

```
sc.ScanInt(num)
IF sc.eot THEN
    *Output empty window message*
ELSE
    largest := num
    sc.ScanInt(num)
    WHILE ~sc.eot DO
        IF num > largest THEN
            largest := num
        END
        sc.ScanInt(num)
    END
    *Output* largest
END
```

Figure 10.7
An algorithm to find the largest value in the focus window.

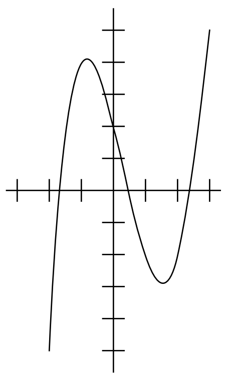
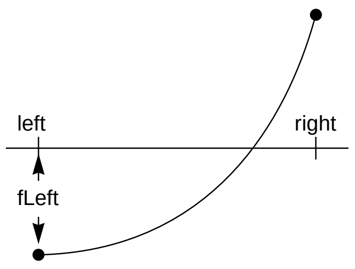
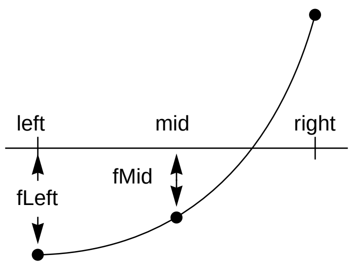


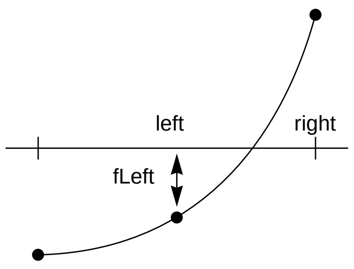
Figure 10.8
A graph of the function
$f(x) = x^3 - x^2 - 4x + 2$



(a) Before the loop executes the first time.

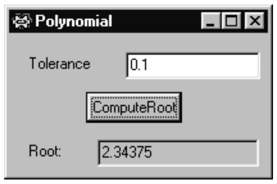


(b) Computation of mid and fMid.



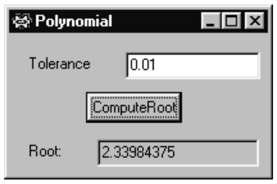
(c) Updating left and fLeft.

Figure 10.9
The bisection algorithm to find a root of *f*(*x*).



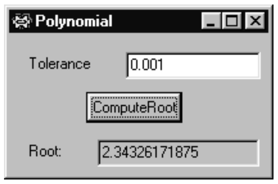
Tolerance 0.1

Root: 2.34375



Tolerance 0.01

Root: 2.33984375



Tolerance 0.001

Root: 2.34326171875

Figure 10.10
Three executions of the bisection algorithm of Listing 10.11.

```
MODULE Pbox10B;
    IMPORT Dialog;
    VAR
        d*: RECORD
            tolerance*: REAL;
            root-: REAL
        END;

    PROCEDURE ComputeRoot*;
        CONST
            a3 = 1.0; a2 = -1.0; a1 = -4.0; a0 = 2.0;
        VAR
            left, fLeft: REAL;
            mid, fMid: REAL;
            right: REAL;
```

Figure 10.11
Computation of the root of a polynomial equation with the bisection algorithm.

```
    BEGIN
        left := 2.0;
        fLeft := ((a3 * left + a2) * left + a1) * left + a0;
        right := 3.0;
        (* Assert: root is between left and right *)
        WHILE ABS(left - right) > d.tolerance DO
            mid := (left + right) / 2.0;
            fMid := ((a3 * mid + a2) * mid + a1) * mid + a0;
            IF fLeft * fMid > 0.0 THEN
                (* Assert: root is between mid and right *)
                left := mid;
                fLeft := fMid
            ELSE
                (* Assert: root is between left and mid *)
                right := mid
            END
        END;
        d.root := (left + right) / 2.0;
        Dialog.Update(d)
    END ComputeRoot;

BEGIN
    d.tolerance := 1.0;
    d.root := 0.0
END Pbox10B.
```

If the focus window contains the text

```
"123-A6002"  35.0  13.00
"123-A6517"  45.0  10.00
"561-B3882"  40.0  12.50
"561-B4559"  40.0  11.00
"561-B7384"  50.0  10.00
```

then the output to the Log should be

```
123-A6002    35.0      455.00
123-A6517    45.0      475.00
561-B3882    40.0      500.00
561-B4559    40.0      440.00
561-B7384    50.0      550.00
Average wages: 484.00
Number with overtime: 2
```

```
VAR
    sc: PboxMappers.Scanner;
    empID: ARRAY 16 OF CHAR;
    hours, rate: REAL;
    wages, totalWages, aveWages: REAL;
    numEmp, numOvertime: INTEGER;
```

```
Initialize variables
Input empID, hours, rate
WHILE ~sc.eot DO
    Process empID, hours, rate
    Input empID, hours, rate
END
Compute the average
Output aveWages, numOvertime
```

```
Initialize variables
Input empID, hours, rate
WHILE ~sc.eot DO
    IF employee did not work overtime THEN
        Compute wages without overtime
    ELSE
        Compute wages with overtime
    END
    Output empID, hours, wages
    Input empID, hours, rate
END
Compute the average
Output aveWages, numOvertime
```

```
totalWages := 0.0
numEmp := 0
numOvertime := 0
Input empID, hours, rate
WHILE ~s.eot DO
    IF employee did not work overtime THEN
        wages := hours * rate
    ELSE
        wages := 40.0 * rate + (hours - 40.0) * 1.5 * rate
        INC(numOvertime)
    END
    totalWages := totalWages + wages
    INC(numEmp)
    Output empID, hours, wages
    Input empID, hours, rate
END
IF numEmp > 0 THEN
    aveWages := totalWages / numEmp
ELSE
    aveWages := 0.00
END;
Output aveWages, numOvertime
```

```
MODULE Pbox10C;
    IMPORT TextModels, TextControllers, PboxMappers, PboxStrings, StdLog;

    PROCEDURE ProcessPayroll*;
        VAR
            md: TextModels.Model;
            cn: TextControllers.Controller;
            sc: PboxMappers.Scanner;
            empID: ARRAY 16 OF CHAR;
            hours, rate: REAL;
            wages, totalWages, aveWages: REAL;
            numEmp, numOvertime: INTEGER;
            outString: ARRAY 32 OF CHAR;
    BEGIN
        cn := TextControllers.Focus();
        IF cn # NIL THEN
            md := cn.text;
            sc.ConnectTo(md);
            totalWages := 0.0; numEmp := 0; numOvertime := 0;
            sc.ScanString(empID); sc.ScanReal(hours); sc.ScanReal(rate);
```

Figure 10.12
A payroll report with summary information

```
            WHILE ~sc.eot DO
                IF hours <= 40 THEN
                    wages := hours * rate
                ELSE
                    wages := 40.0 * rate + (hours - 40.0) * 1.5 * rate;
                    INC(numOvertime)
                END;
                StdLog.String(empID);
                PboxStrings.RealToString(hours, 8, 1, outString); StdLog.String(outString);
                PboxStrings.RealToString(wages, 12, 2, outString); StdLog.String(outString);
                StdLog.Ln;
                totalWages := totalWages + wages;
                INC(numEmp);
                sc.ScanString(empID); sc.ScanReal(hours); sc.ScanReal(rate)
            END;
            IF numEmp > 0 THEN
                aveWages := totalWages / numEmp
            ELSE
                aveWages := 0.00
            END;
            StdLog.String("Average wages: ");
            PboxStrings.RealToString(aveWages, 1, 2, outString); StdLog.String(outString); StdLog.Ln;
            StdLog.String("Number with overtime: ");
            PboxStrings.IntToString(numOvertime, 1, outString); StdLog.String(outString); StdLog.Ln;
        END
    END ProcessPayroll;

END Pbox10C.
```

The structured programming theorem

Any algorithm, no matter how large or complicated, can be written with only three control statements—sequence, which is one statement following another, the IF statement, and the WHILE statement.

```
sum := 0
i := 1
WHILE i <= 100 DO
    sum := sum + i
    INC(i)
END
Output sum
```

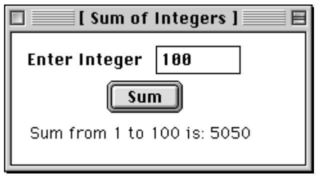
Figure 10.13
An algorithm for the sum of consecutive integers with a WHILE loop.

```
MODULE Pbox10D;
    IMPORT Dialog, PboxStrings;
    VAR
        d*: RECORD
            num*: INTEGER;
            message-: ARRAY 64 OF CHAR
        END;

    PROCEDURE ComputeSum*;
        VAR
            sum, i: INTEGER;
            intString: ARRAY 16 OF CHAR;
    BEGIN
        sum := 0;
        FOR i := 1 TO d.num DO
            sum := sum + i
        END;
        PboxStrings.IntToString(d.num, 1, intString);
        d.message := "Sum from 1 to " + intString + " is: ";
        PboxStrings.IntToString(sum, 1, intString);
        d.message := d.message + intString;
        Dialog.Update(d)
    END ComputeSum;

BEGIN
    d.num := 0;
    d.message := ""
END Pbox10D.
```

Figure 10.14
Computing the sum of the first d.num integers with a FOR loop.



```
100
Sum from 1 to 100 is: 5050
```

Figure 10.15
The dialog box for the program of Figure 10.14.

```
sum := d.num * (d.num + 1) / 2
Output sum
```

Figure 10.16
A better algorithm for the sum of consecutive integers.

```
WHILE C1 DO
    S1
END
```

is written in GCL as

do $C1 \rightarrow S1$ **od**

```
$s$ := 0.0; $nA$ := 0; sc.ScanR($b$);
do ¬sc.eot → $s$ := $s + b$; $nA$ := $nA + 1$; sc.ScanR($b$) od
if  $nA > 0$ → Output $s / nA$
 [] $nA \le 0$ → Output a no accounts message
fi
```