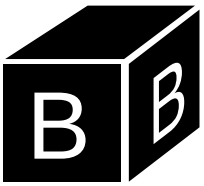


Chapter *8*

Nested Selections

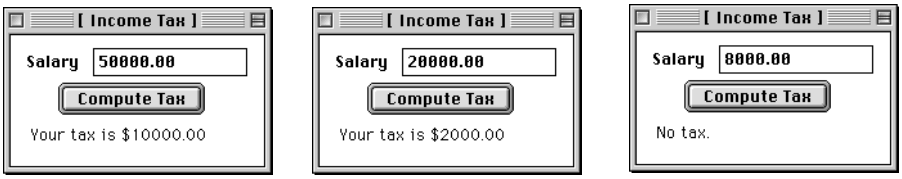


Figure 8.1
A dialog box that requires more than two alternative computations.

```
MODULE Pbox08A;
    IMPORT Dialog, PboxStrings;
    VAR
        d*: RECORD
            salary*: REAL;
            message-: ARRAY 64 OF CHAR
        END;

    PROCEDURE IncomeTax*;
        CONST
            lowRate = 0.20;
            highRate = 0.30;
            minTaxable = 10000.00;
            maxTaxable = 30000.00;
        VAR
            tax: REAL;
            taxString: ARRAY 32 OF CHAR;
    BEGIN
        IF d.salary > minTaxable THEN
            IF d.salary <= maxTaxable THEN
                tax := (d.salary - minTaxable) * lowRate
            ELSE
                tax := (maxTaxable - minTaxable) * lowRate + (d.salary - maxTaxable) * highRate
            END;
            PboxStrings.RealToString(tax, 1, 2, taxString);
            d.message := "Your tax is $" + taxString
        ELSE
            d.message := "No tax."
        END;
        Dialog.Update(d)
    END IncomeTax;

BEGIN
    d.salary := 0.0;
    d.message := ""
END Pbox08A.
```

Figure 8.2
An income tax computation with a nested IF statement.

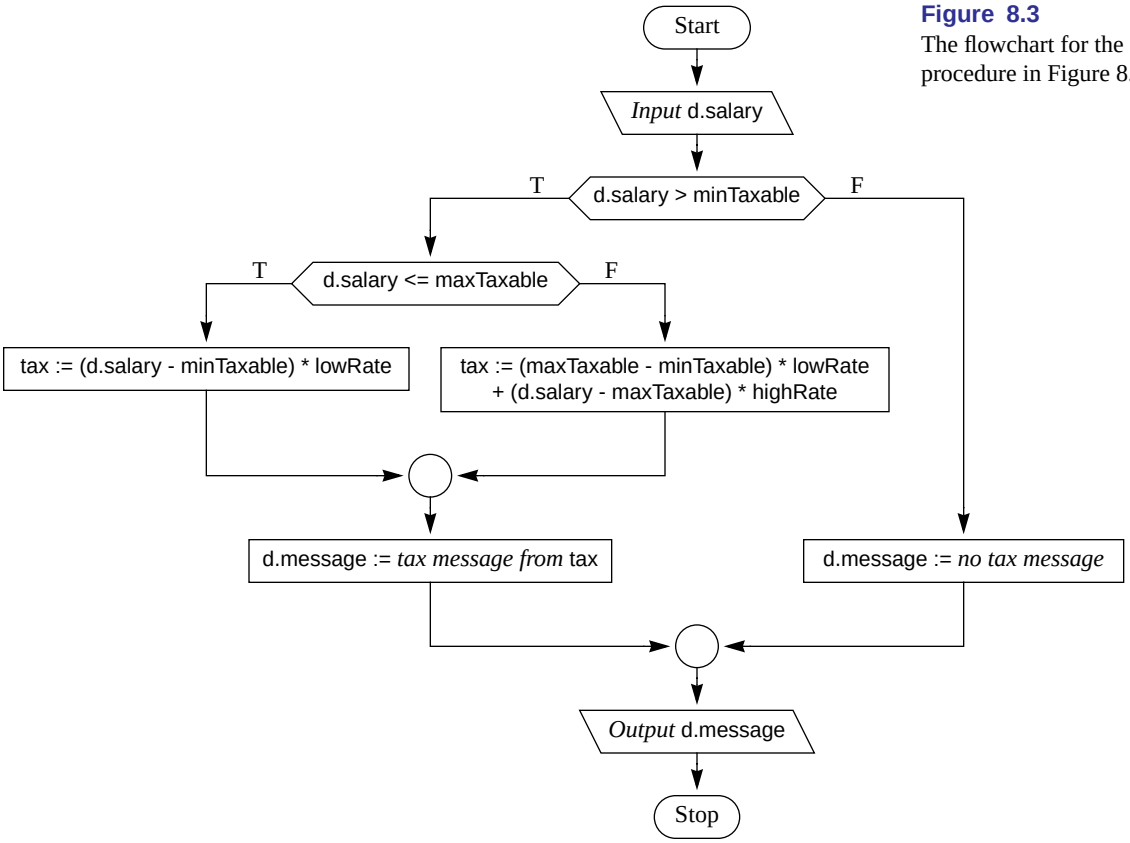


Figure 8.3
The flowchart for the procedure in Figure 8.2.

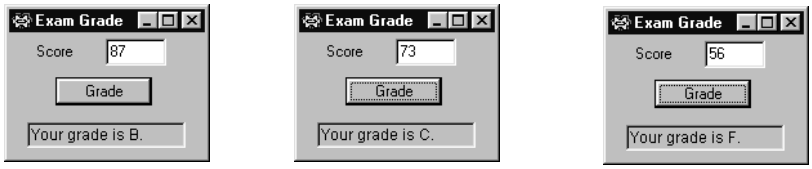


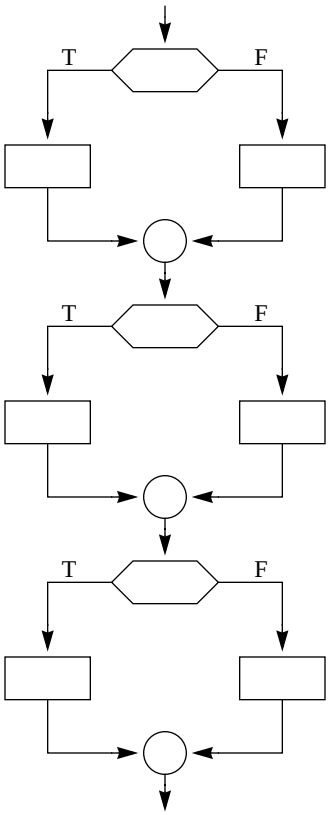
Figure 8.4
A dialog box for computing a letter grade from an exam score.

```
MODULE Pbox08B;
    IMPORT Dialog;
    VAR
        d*: RECORD
            score*: INTEGER;
            message-: ARRAY 64 OF CHAR
        END;

    PROCEDURE TestGrade*;
    BEGIN
        IF d.score >= 90 THEN
            d.message := "Your grade is A."
        ELSE
            IF d.score >= 80 THEN
                d.message := "Your grade is B."
            ELSE
                IF d.score >= 70 THEN
                    d.message := "Your grade is C."
                ELSE
                    IF d.score >= 60 THEN
                        d.message := "Your grade is D."
                    ELSE
                        d.message := "Your grade is F."
                    END
                END
            END
        END;
        Dialog.Update(d)
    END TestGrade;

BEGIN
    d.score := 0;
    d.message := ""
END Pbox08B.
```

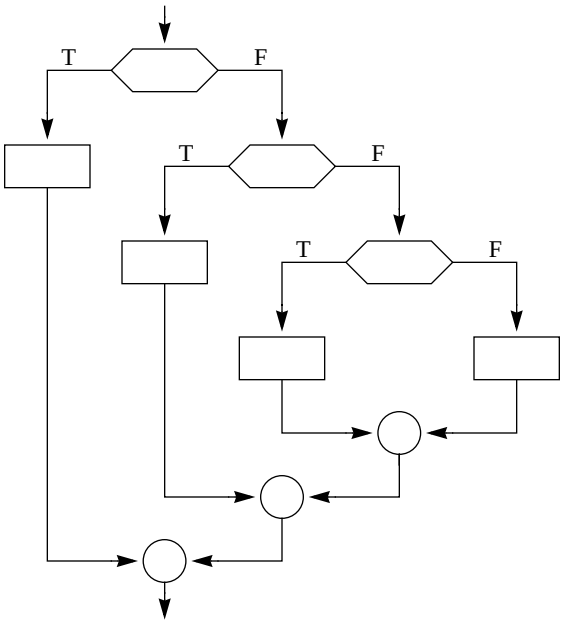
Figure 8.5
Conversion of an integer exam score into a letter grade.

```
PROCEDURE TestGrade*;
BEGIN
    IF d.score >= 90 THEN
        d.message := "Your grade is A."
    ELSIF d.score >= 80 THEN
        d.message := "Your grade is B."
    ELSIF d.score >= 70 THEN
        d.message := "Your grade is C."
    ELSIF d.score >= 60 THEN
        d.message := "Your grade is D."
    ELSE
        d.message := "Your grade is F."
    END;
    Dialog.Update(d)
END TestGrade;
```

```
IF d.score >= 90 THEN
    d.message := "Your grade is A."
END;
IF d.score >= 80 THEN
    d.message := "Your grade is B."
END;
IF d.score >= 70 THEN
    d.message := "Your grade is C."
END;
IF d.score >= 60 THEN
    d.message := "Your grade is D.";
ELSE
    d.message := "Your grade is F.";
END
```



(a) Sequential IF statements



(b) Nested IF statements

Figure 8.6
Flowcharts for sequential IF statements versus a single IF statement with two ELSIF parts.

```
IF d.salary > minTaxable THEN
    IF d.salary <= maxTaxable THEN
        ASSERT(
        tax := (d.salary - minTaxable) * lowRate
    ELSE
        ASSERT(
        tax := (maxTaxable - minTaxable) * lowRate + (d.salary - maxTaxable) * highRate
    END;
    d.message := tax message from tax
ELSE
    ASSERT(
    d.message := no tax message
END;
```

```
IF d.salary > minTaxable THEN
    IF d.salary <= maxTaxable THEN
        ASSERT((minTaxable < d.salary) & (d.salary <= maxTaxable), 100);
        tax := (d.salary - minTaxable) * lowRate
    ELSE
        ASSERT(
        tax := (maxTaxable - minTaxable) * lowRate + (d.salary - maxTaxable) * highRate
    END;
    d.message := tax message from tax
ELSE
    ASSERT(
    d.message := no tax message
END;
```

```
IF d.salary > minTaxable THEN
    IF d.salary <= maxTaxable THEN
        ASSERT((minTaxable < d.salary) & (d.salary <= maxTaxable), 100);
        tax := (d.salary - minTaxable) * lowRate
    ELSE
        ASSERT(d.salary > maxTaxable, 101);
        tax := (maxTaxable - minTaxable) * lowRate + (d.salary - maxTaxable) * highRate
    END;
    d.message := tax message from tax
ELSE
    ASSERT(
    d.message := no tax message
END;
```

```
IF d.salary > minTaxable THEN
    IF d.salary <= maxTaxable THEN
        ASSERT((minTaxable < d.salary) & (d.salary <= maxTaxable), 100);
        tax := (d.salary - minTaxable) * lowRate
    ELSE
        ASSERT(d.salary > maxTaxable, 101);
        tax := (maxTaxable - minTaxable) * lowRate + (d.salary - maxTaxable) * highRate
    END;
    d.message := tax message from tax
ELSE
    ASSERT(d.salary <= minTaxable, 102);
    d.message := no tax message
END;
```

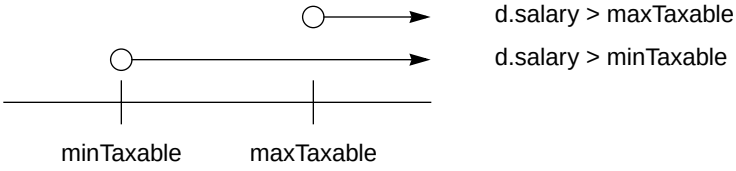


Figure 8.7
The real number line showing the conditions d.salary > maxTaxable and d.salary > minTaxable.

```
IF age > 65 THEN
    ASSERT(
    Statement 1
ELSIF age > 21 THEN
    ASSERT(
    Statement 2
ELSE
    ASSERT(
    Statement 3
END
```

```
IF age > 65 THEN
    ASSERT(age > 65, 100);
    Statement 1
ELSIF age > 21 THEN
    ASSERT(
    Statement 2
ELSE
    ASSERT(
    Statement 3
END
```

```
IF age > 65 THEN
    ASSERT(age > 65, 100);
    Statement 1
ELSIF age > 21 THEN
    ASSERT((21 < age) & (age <= 65), 101);
    Statement 2
ELSE
    ASSERT(
    Statement 3
END
```

```
IF age > 65 THEN
    ASSERT(age > 65, 100);
    Statement 1
ELSIF age > 21 THEN
    ASSERT((21 < age) & (age <= 65), 101);
    Statement 2
ELSE
    ASSERT(age <= 21, 102);
    Statement 3
END
```

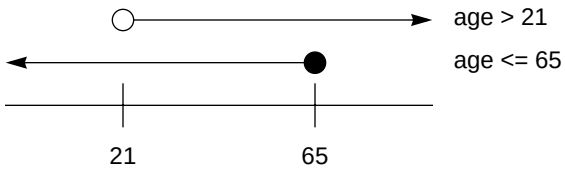


Figure 8.8
The number line showing the two conditions age > 21 and age <= 65.

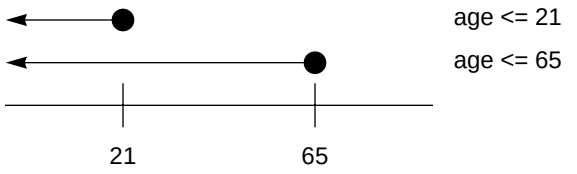


Figure 8.9
The number line showing the two conditions age <= 21 and age <= 65.

```
IF quantity < 200 THEN
    Statement 1
ELSIF quantity >= 100 THEN
    Statement 2
ELSE
    Statement 3
END
```

```
IF quantity < 200 THEN
    ASSERT(quantity < 200, 100);
    Statement 1
ELSIF quantity >= 100 THEN

    Statement 2
ELSE

    Statement 3
END
```

```
IF quantity < 200 THEN
    ASSERT(quantity < 200, 100);
    Statement 1
ELSIF quantity >= 100 THEN
    ASSERT(quantity >= 200, 101);
    Statement 2
ELSE

    Statement 3
END
```

```
IF quantity < 200 THEN
    ASSERT(quantity < 200, 100);
    Statement 1
ELSIF quantity >= 100 THEN
    ASSERT(quantity >= 200, 101);
    Statement 2
ELSE
    HALT(102);
    Statement 3
END
```

```
IF (15 <= n) & (n < 20) THEN
   IF (n > 10) THEN

      Statement 1
   ELSE

      Statement 2
   END
ELSE

   Statement 3
END
```

```
IF (15 <= n) & (n < 20) THEN
   IF (n > 10) THEN
      ASSERT((15 <= n) & (n < 20), 100);
      Statement 1
   ELSE

      Statement 2
   END
ELSE

   Statement 3
END
```

```
IF (15 <= n) & (n < 20) THEN
    IF (n > 10) THEN
        ASSERT((15 <= n) & (n < 20), 100);
        Statement 1
    ELSE
        HALT(101);
        Statement 2
    END
ELSE

    Statement 3
END
```

```
IF (15 <= n) & (n < 20) THEN
    IF (n > 10) THEN
        ASSERT((15 <= n) & (n < 20), 100);
        Statement 1
    ELSE
        HALT(101);
        Statement 2
    END
ELSE
    ASSERT((n < 15) OR (n >= 20), 102);
    Statement 3
END
```

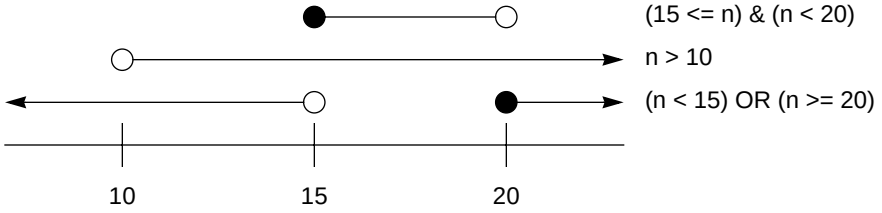


Figure 8.10
The number line for Example 8.4.

```
IF weight > 150.0 THEN
    Statement 1
END;
IF (weight > 50.0) & (weight <= 150.0) THEN
    Statement 2
END;
IF (weight <= 50.0) THEN
    Statement 3
END
```

```
IF weight > 150.0 THEN
    Statement 1
ELSIF weight > 50.0 THEN
    Statement 2
ELSE
    Statement 3
END
```

```
IF price > 2000 THEN
    Statement 1
ELSIF price > 1000 THEN
    Statement 2
ELSIF price <= 1000 THEN
    Statement 3
END
```

```
IF price > 2000 THEN
    Statement 1
ELSIF price > 1000 THEN
    Statement 2
ELSE
    Statement 3
END
```

if $B1 \rightarrow S1$
[] $B2 \rightarrow S2$
[] $B3 \rightarrow S3$
[] $B4 \rightarrow S4$
fi

if $x > y \rightarrow x, y := y, x$
fi

```
IF price > 2000 THEN
    Statement 1
ELSIF price > 1000 THEN
    Statement 2
ELSE
    Statement 3
END
```

if $price > 2000 \rightarrow S1$
[] $price > 1000 \rightarrow S2$
[] $price \leq 1000 \rightarrow S3$
fi

if $price > 2000 \rightarrow S1$
[] $1000 < price \leq 2000 \rightarrow S2$
[] $price \leq 1000 \rightarrow S3$
fi