```
Function RaisedSalary(ByVal salary As Double) As Double
  If salary <= 40000 Then
    Return 1.05 * salary
  Else
    Return salary + 2000 + 0.02 * (salary — 40000)
  End If
End Function

Sub DisplayOutput(ByVal firstName As String, ByVal lastName As String,
                  ByVal newSalary As Double)
  txtOutput.Text = "New salary for " & firstName & " " & lastName &
                   " is " & FormatCurrency(newSalary) & "."
End Sub
```

13.

```
Private Sub btnCalculate_Click(...) Handles btnCalculate.Click
  Dim annualRateOfInterest, monthlyPayment, begBalance As Double
  Dim intForMonth, redOfPrincipal, endBalance As Double
  InputData(annualRateOfInterest, monthlyPayment, begBalance)
  Calculate(annualRateOfInterest, monthlyPayment, begBalance,
            intForMonth, redOfPrincipal, endBalance)
  DisplayData(intForMonth, redOfPrincipal, endBalance)
End Sub

Sub InputData(ByRef annualRateOfInterest As Double,
              ByRef monthlyPayment As Double,
              ByRef begBalance As Double)
  annualRateOfInterest = CDbl(txtAnnualRateOfInterest.Text)
  monthlyPayment = CDbl(txtMonthlyPayment.Text)
  begBalance = CDbl(txtBegBalance.Text)
End Sub

Sub Calculate(ByVal annualRateOfInterest As Double,
              ByVal monthlyPayment As Double,
              ByVal begBalance As Double, ByRef intForMonth As Double,
              ByRef redOfPrincipal As Double, ByRef endBalance As Double)
  Dim monthlyRateOfInterest As Double = annualRateOfInterest / 12
  intForMonth = (monthlyRateOfInterest / 100) * begBalance
  redOfPrincipal = monthlyPayment — intForMonth
  endBalance = begBalance — redOfPrincipal
End Sub

Sub DisplayData(ByVal intForMonth, ByVal redOfPrincipal,
                ByVal endBalance)
  txtIntForMonth.Text = FormatCurrency(intForMonth)
  txtRedOfPrincipal.Text = FormatCurrency(redOfPrincipal)
  txtEndBalance.Text = FormatCurrency(endBalance)
End Sub
```

Chapter 6

Exercises 6.1

1. `18`   **3.** `10`   **5.** `Maximum number: 7`

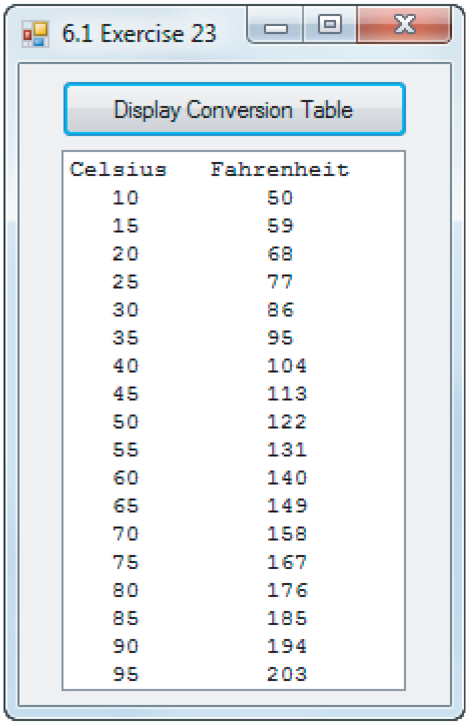
7. Infinite loop. (To end the program, click on the *Stop Debugging* button on the Toolbar.)

9. Do and Loop are interchanged   **11.** `While num >= 7`   **13.** `Until response <> "Y"`

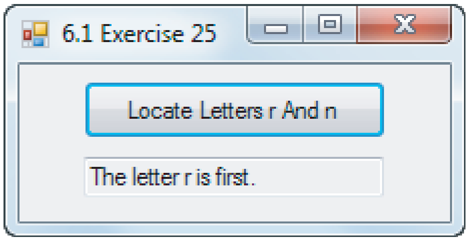
15. `Until name = ""`   17. `Until (a <= 1) Or (a >= 3)`   19. `While n = 0`

21.
```
Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  'Request and display three names.
  Dim name As String, num As Integer = 0
  Do While num < 3
    name = InputBox("Enter a name:")
    lstOutput.Items.Add(name)
    num +=1     'Add 1 to value of num.
  Loop
End Sub
```

23.
```
Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  Dim celsius As Double = 10
  lstOutput.Items.Add("Celsius   Fahrenheit")
  Do While celsius <= 95
    lstOutput.Items.Add("   " & celsius & "         " &
                        Fahrenheit(celsius))
    celsius += 5
  Loop
End Sub

Function Fahrenheit(ByVal celsius As Double) As Double
  'Convert Celsius to Fahrenheit
  Return (9 / 5) * celsius + 32
End Function
```



```
Celsius   Fahrenheit
   10          50
   15          59
   20          68
   25          77
   30          86
   35          95
   40          104
   45          113
   50          122
   55          131
   60          140
   65          149
   70          158
   75          167
   80          176
   85          185
   90          194
   95          203
```

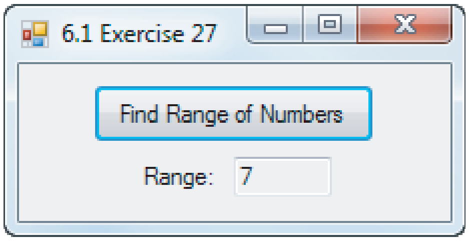


The letter r is first.

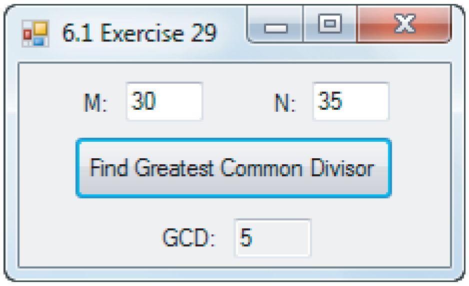
25.
```
Private Sub btnLocate_Click(...) Handles btnLocate.Click
  Dim word As String = ""
  Dim rPlace, nPlace As Integer
  Do
    InputWord(word)
    rPlace = word.IndexOf("r")
    nPlace = word.IndexOf("n")
    If (rPlace = -1) Or (nPlace = -1) Then
      MessageBox.Show("That word does not contain both r and n.", "")
    End If
  Loop Until (rPlace > -1) And (nPlace > -1)
  ShowFirst(rPlace, nPlace)
End Sub

Sub InputWord(ByRef word As String)
  Dim prompt As String
  prompt = "Enter a word containing the letters 'r' and 'n'."
  word = InputBox(prompt, "Enter Word")
End Sub

Sub ShowFirst(ByVal rPlace As Integer, ByVal nPlace As Integer)
  'Tell which letter, r or n, comes first.
  If nPlace > rPlace Then
    txtOutput.Text = "The letter r is first."
  Else
    txtOutput.Text = "The letter n is first."
  End If
End Sub
```

27.

```
Private Sub btnCompute_Click(...) Handles btnCompute.Click
  Dim num, max, min As Double
  Dim count As Double = 0
  Dim prompt As String = "Enter a nonnegative number. " &
                         "Enter -1 to terminate entering numbers."
  num = CDbl(InputBox(prompt))
  max = num
  min = num
  Do While num >= 0
    count += 1
    num = CDbl(InputBox(prompt))
    If (num <> -1) Then
      If num < min Then
        min = num
      End If
      If num > max Then
        max = num
      End If
    End If
  Loop
  If count > 0 Then
    txtRange.Text = CStr(max - min)
  Else
    MessageBox.Show("No numbers were entered.")
  End If
End Sub
```



Range: 7



M: 30 N: 35

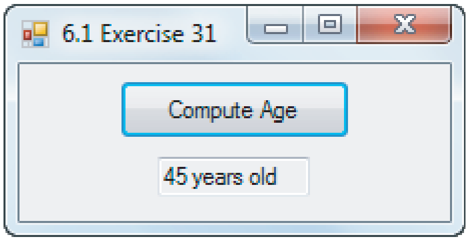
GCD: 5

29.

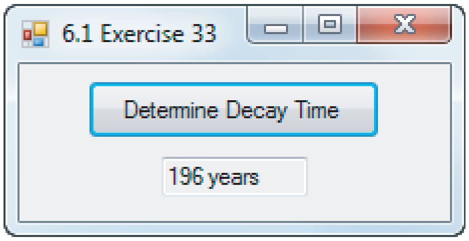
```
Private Sub btnFind_Click(...) Handles btnFind.Click
  Dim m, n, t As Integer
  InputIntegers(m, n)
  Do While n <> 0
    t = n
    n = m Mod n     'Remainder after m is divided by n
    m = t
  Loop
  txtOutput.Text = CStr(m)
End Sub

Sub InputIntegers(ByRef m As Integer, ByRef n As Integer)
  m = CInt(txtM.Text)
  n = CInt(txtN.Text)
End Sub
```

31.

```
Private Sub btnCompute_Click(...) Handles btnCompute.Click
  Dim age As Integer = 1
  Do While 1980 + age <> age ^ 2
    age += 1
  Loop
  txtOutput.Text = age & " years old"
End Sub
```



```
45 years old
```



```
196 years
```

33.

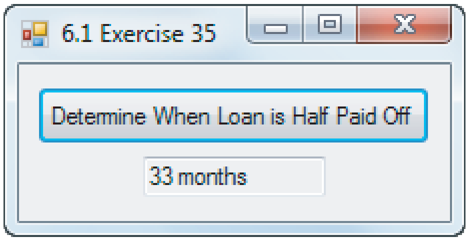
```
Private Sub btnDetermine_Click(...) Handles btnDetermine.Click
  Dim amount As Double = 100
  Dim yrs As Integer = 0
  Do Until amount < 1
    amount = 0.5 * amount
    yrs += 28
  Loop
  txtOutput.Text = yrs & " years"
End Sub
```

35.

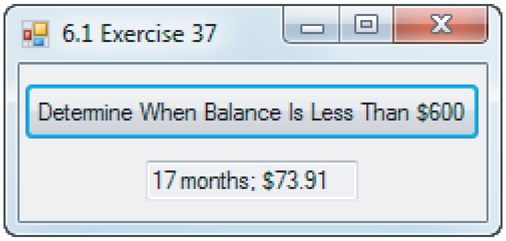
```
Private Sub btnDetermine_Click(...) Handles btnDetermine.Click
  Const INTEREST_PER_MONTH As Double = 0.005
  Dim loanAmount As Double = 15000
  Dim months As Integer = 0
  Dim balance As Double = loanAmount
  Do Until balance < loanAmount / 2
    balance = (1 + INTEREST_PER_MONTH) * balance – 290
    months += 1
  Loop
  txtOutput.Text = months & " months"
End Sub
```

37.

```
Private Sub btnDetermine_Click(...) Handles btnDetermine.Click
  Dim months As Integer = 0
  Dim balance As Double = 10000
  Do Until balance < 600
    balance = 1.003 * balance – 600
    months += 1
  Loop
  txtOutput.Text = months & " months; " & FormatCurrency(balance)
End Sub
```



33 months



17 months; $73.91

Exercises 6.2

1.
```
Pass #1
Pass #2
Pass #3
Pass #4
```

3.
```
2
4
6
8
Who do we appreciate?
```

5.
```
5
6
7
```

7. ●●●●●●●●●●

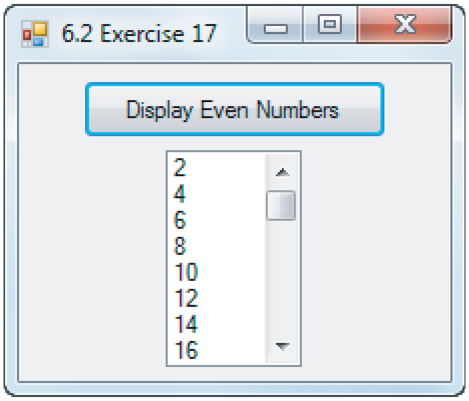
9. `4`

11. The loop is never executed since 25 is greater than 1 and the step is negative.

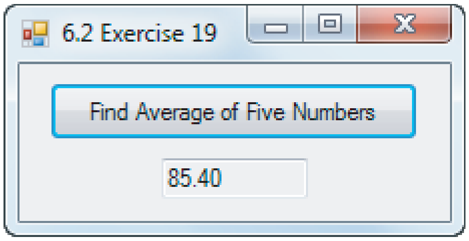
13. The For … Next loop will not execute since 20 is greater than 0. You must add `Step -1` to the end of the For statement.

15.
```
Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  For num As Integer = 1 To 9 Step 2
    lstBox.Items.Add(num)
  Next
End Sub
```

17.
```
Private Sub btndisplay_Click(...) Handles btndisplay.Click
  For i As Integer = 2 To 100 Step 2
    lstOutput.Items.Add(i)
  Next
End Sub
```



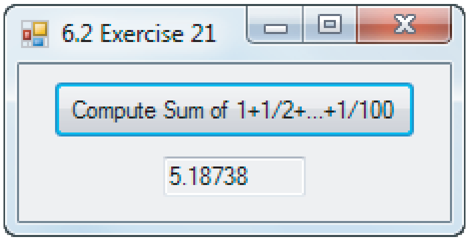
```
2
4
6
8
10
12
14
16
```



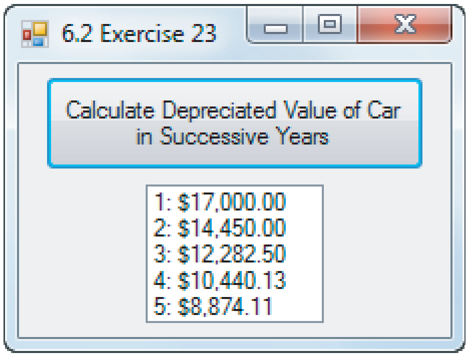
85.40

19.
```
Private Sub btnFind_Click(...) Handles btnFind.Click
  Dim sum As Double = 0, num as Double = 0
  For i As Integer = 1 To 5
    num = CDbl(InputBox("Enter #" & i))
    sum += num
  Next
  txtAverage.Text = FormatNumber(sum / 5, 2)
End Sub
```

21.
```
Private Sub btnCompute_Click(...) Handles btnCompute.Click
  Dim sum As Double = 0
  For denominator As Double = 1 To 100
    sum += 1 / denominator
  Next
  txtOutput.Text = FormatNumber(sum, 5)
End Sub
```



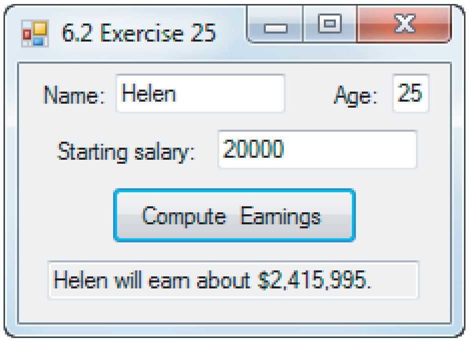
```
5.18738
```



```
1: $17,000.00
2: $14,450.00
3: $12,282.50
4: $10,440.13
5: $8,874.11
```

23.
```
Private Sub btnCalculate_Click(...) Handles btnCalculate.Click
  Dim value As Double = 20000
  For i As Integer = 1 To 5
    value = 0.85 * value
    lstOutput.Items.Add(i & ": " & FormatCurrency(value))
  Next
End Sub
```

25.
```
Private Sub btnCompute_Click(...) Handles btnCompute.Click
  Dim PERCENT_RAISE As Double = 0.05
  Dim name As String, age As Integer, salary As Double
  Dim earnings As Double = 0
  name = txtName.Text
  age = CInt(txtAge.Text)
  salary = CDbl(txtSalary.Text)
  For i As Integer = age To 64
    earnings += salary
    salary = salary + (PERCENT_RAISE * salary)
  Next
  txtOutput.Text = name & " will earn about " &
                   FormatCurrency(earnings, 0) & "."
End Sub
```



Name: Helen Age: 25

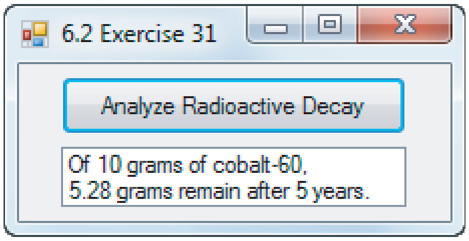
Starting salary: 20000

```
Helen will earn about $2,415,995.
```

27.
```
Private Sub btnComputeIdealWeights_Click(...) Handles _
                   btnComputeIdealWeights.Click
  Dim lower, upper As Integer
  lstWeightTable.Items.Clear()
  InputBounds(lower, upper)
  ShowWeights(lower, upper)
End Sub

Function IdealMan(ByVal height As Integer) As Double
  'Compute the ideal weight of a man given his height
  Return 4 * height – 128
End Function

Function IdealWoman(ByVal height As Integer) As Double
  'Compute the ideal weight of a woman given her height
  Return 3.5 * height – 108
End Function

Sub InputBounds(ByRef lower As Integer, ByRef upper As Integer)
  lower = CInt(InputBox("Enter lower bound on height in inches:"))
  upper = CInt(InputBox("Enter upper bound on height in inches:"))
End Sub

Sub ShowWeights(ByVal lower As Integer, ByVal upper As Integer)
  lstWeightTable.Items.Add("        " & "WEIGHT  " & " " & "WEIGHT")
  lstWeightTable.Items.Add("HEIGHT" & "  " & "WOMEN   " & " " & "MEN")
  For height As Integer = lower To upper
    lstWeightTable.Items.Add(height & "      " &
              FormatNumber(IdealWoman(height), 1) &
              "    " & FormatNumber(IdealMan(height), 1))
  Next
End Sub
```

29.
```
Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  Dim balance As Double = 0
  Dim yr As Integer = 2010
  For i As Integer = 1 To 120
    balance = (1.0025) * balance + 100
    If i Mod 12 = 0 Then
      lstOutput.Items.Add(yr & "  " & FormatCurrency(balance))
      yr += 1
    End If
  Next
End Sub
```

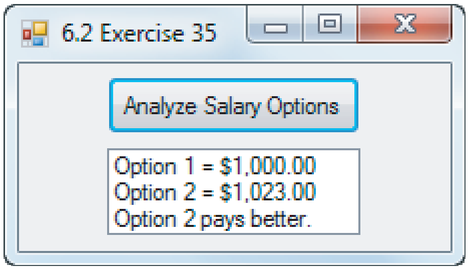
31.
```
Private Sub btnAnalyze_Click(...) Handles btnAnalyze.Click
  Const DECAY_RATE As Double = 0.12
  Dim grams As Double
  grams = 10
  For yearNum As Integer = 1 To 5
    grams = (1 – DECAY_RATE) * grams
  Next
  lstOutput.Items.Add("Beginning with 10 grams of cobalt 60,")
  lstOutput.Items.Add(FormatNumber(grams) &
                      " grams remain after 5 years.")
End Sub
```



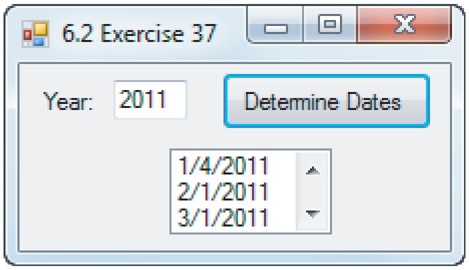
```
Of 10 grams of cobalt-60,
5.28 grams remain after 5 years.
```

33.
```
Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  Dim price, quantity As Double
  lstOutput.Items.Clear()
  quantity = 80  'current crop of soybeans in millions of bushels
  lstOutput.Items.Add("YEAR" & "    " & "QUANTITY" & "    " & "PRICE")
  For yr As Integer = 2010 To 2020
    price = 20 — 0.1 * quantity
    lstOutput.Items.Add(yr & "     " & FormatNumber(quantity) &
                        "     " & FormatCurrency(price))
    quantity = 5 * price — 10
  Next
End Sub
```

35.
```
Private Sub btnAnalyzeOptions_Click(...) Handles btnAnalyzeOptions.Click
  'Compare salaries
  Dim opt1, opt2 As Double
  opt1 = Option1()
  opt2 = Option2()
  lstOutput.Items.Add("Option 1 = " & FormatCurrency(opt1))
  lstOutput.Items.Add("Option 2 = " & FormatCurrency(opt2))
  If opt1 > opt2 Then
    lstOutput.Items.Add("Option 1 pays better.")
  ElseIf opt1 = opt2 Then
    lstOutput.Items.Add("Options pay the same.")
  Else
    lstOutput.Items.Add("Option 2 pays better.")
  End If
End Sub

Function Option1() As Double
  'Compute total salary with a flat salary of $100/day
  Dim sum As Integer = 0
  For i As Integer = 1 To 10
    sum += 100
  Next
  Return sum
End Function

Function Option2() As Double
  'Compute total salary starting at $1 and doubling each day
  Dim sum As Integer = 0, daySalary As Integer = 1
  For i As Integer = 1 To 10
    sum += daySalary
    daySalary = 2 * daySalary
  Next
  Return sum
End Function
```



```
Option 1 = $1,000.00
Option 2 = $1,023.00
Option 2 pays better.
```



Year: 2011

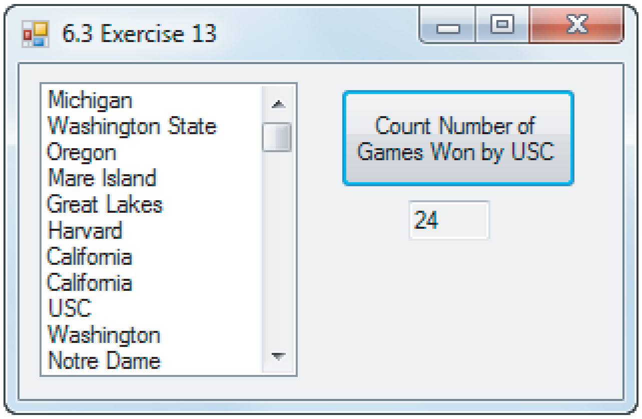
```
1/4/2011
2/1/2011
3/1/2011
```

37.
```
Private Sub btnDetermine_Click(...) Handles btnDetermine.Click
  Dim dt As Date = CDate("#1/1/" & mtbYear.Text & "#")
  Dim d As Date
  For i As Integer = 0 To 11
    d = dt.AddMonths(i)
    lstOutput.Items.Add(FirstTuesday(d))
  Next
End Sub

Function FirstTuesday(ByVal d As Date) As Date
  For i As Integer = 0 To 6
    If FormatDateTime(d.AddDays(i),
                       DateFormat.LongDate).StartsWith("Tuesday") Then
      Return d.AddDays(i)
    End If
  Next
End Function
```

Exercises 6.3

1. `Mozart` **3.** `Tchaikovsky` **5.** `3` **7.** `80` **9.** `70` **11.** `300`

13.
```
Private Sub btnCount_Click(...) Handles btnCount.Click
  Dim numWon As Integer = 0
  For i As Integer = 0 To lstBox.Items.Count − 1
    If CStr(lstBox.Items(i)) = "USC" Then
      numWon += 1
    End If
  Next
  txtOutput.Text = CStr(numWon)
End Sub
```



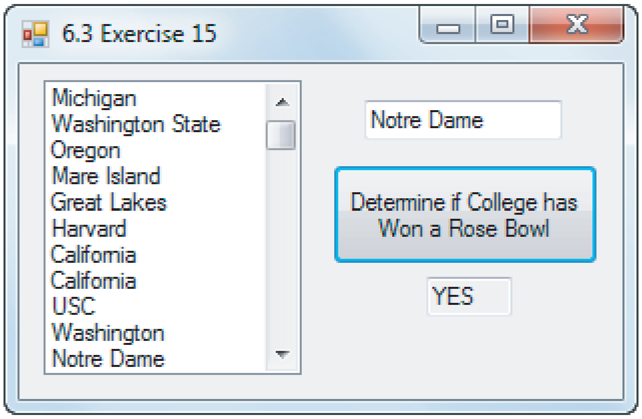
```
Michigan
Washington State
Oregon
Mare Island
Great Lakes
Harvard
California
California
USC
Washington
Notre Dame
```

```
24
```

```
15. Private Sub btnCount_Click(...) Handles btnDetermine.Click
      Dim college As String = txtCollege.Text
      txtOutput.Clear()
      For i As Integer = 0 To lstBox.Items.Count - 1
        If CStr(lstBox.Items(i)) = college Then
          txtOutput.Text = "YES"
          Exit For
        End If
      Next
      If txtOutput.Text = "" Then
        txtOutput.Text = "NO"
      End If
    End Sub
```

or

```
    Private Sub btnCount_Click(...) Handles btnDetermine.Click
      Dim college As String = txtCollege.Text
      Dim i As Integer = 0
      Dim found As Boolean = False
      Do Until (found = True) Or (i = lstBox.Items.Count)
        If CStr(lstBox.Items(i)) = college Then
          found = True
        End If
        i += 1
      Loop
      If found Then
        txtOutput.Text = "YES"
      Else
        txtOutput.Text = "NO"
      End If
    End Sub
```



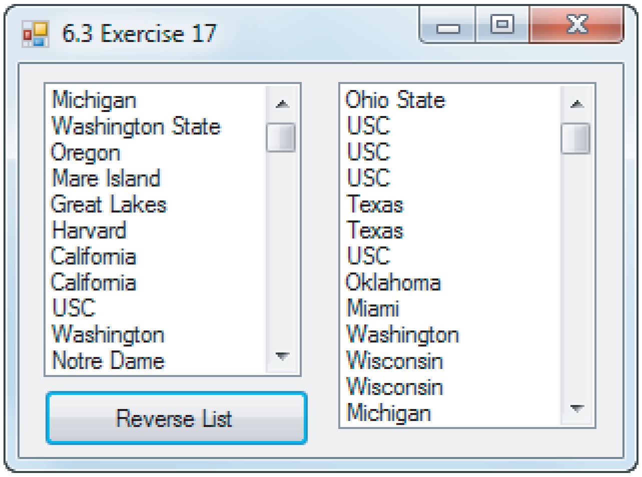
Michigan
Washington State
Oregon
Mare Island
Great Lakes
Harvard
California
California
USC
Washington
Notre Dame

Notre Dame

YES

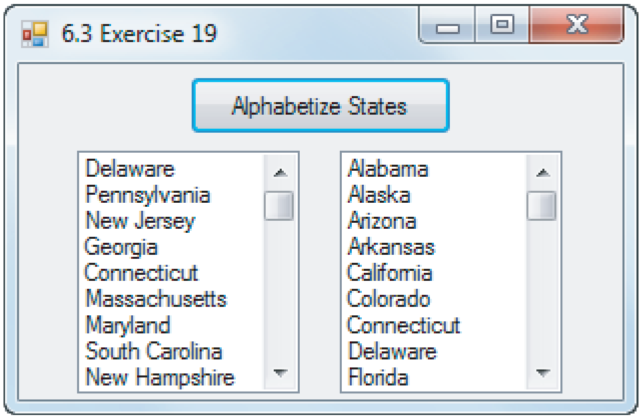
```
17. Private Sub btnReverse_Click(...) Handles btnReverse.Click
      Dim highestIndex As Integer = lstBox.Items.Count - 1
      For i As Integer = highestIndex To 0 Step -1
        lstBox2.Items.Add(lstBox.Items(i))
      Next
    End Sub
```

```
19. Private Sub btnAlphabetize_Click(...) Handles btnAlphabetize.Click
      lstBox2.Sorted = True
      Dim highestIndex As Integer = lstBox.Items.Count - 1
      For i As Integer = 0 To highestIndex
        lstBox2.Items.Add(lstBox.Items(i))
      Next
    End Sub
```



Michigan
Washington State
Oregon
Mare Island
Great Lakes
Harvard
California
California
USC
Washington
Notre Dame

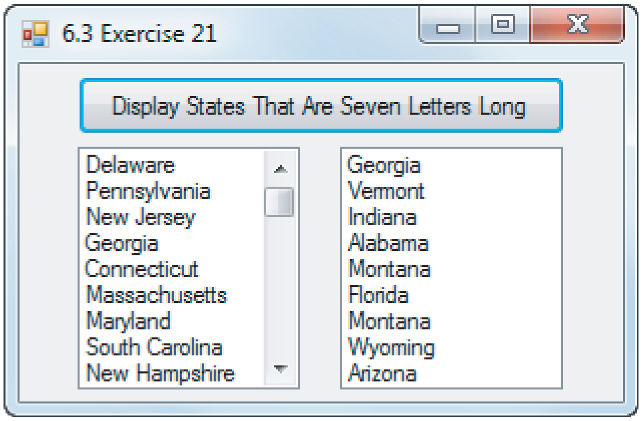
Ohio State
USC
USC
USC
Texas
Texas
USC
Oklahoma
Miami
Washington
Wisconsin
Wisconsin
Michigan



Delaware
Pennsylvania
New Jersey
Georgia
Connecticut
Massachusetts
Maryland
South Carolina
New Hampshire

Alabama
Alaska
Arizona
Arkansas
California
Colorado
Connecticut
Delaware
Florida

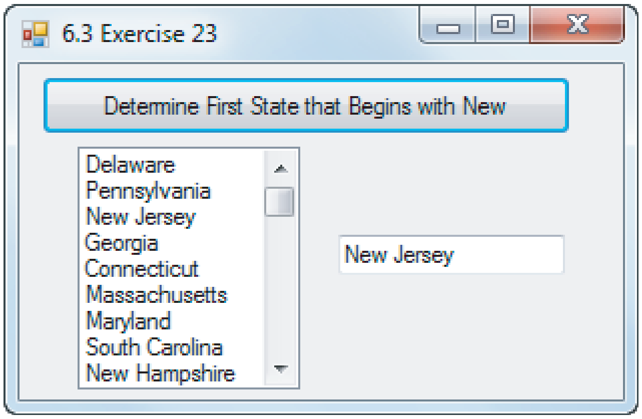
21.
```
Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  Dim highestIndex As Integer = lstBox.Items.Count – 1
  Dim state As String
  For i As Integer = 0 To highestIndex
    state = CStr(lstBox.Items(i))
    If state.Length = 7 Then
      lstBox2.Items.Add(state)
    End If
  Next
End Sub
```



Delaware
Pennsylvania
New Jersey
Georgia
Connecticut
Massachusetts
Maryland
South Carolina
New Hampshire

Georgia
Vermont
Indiana
Alabama
Montana
Florida
Montana
Wyoming
Arizona

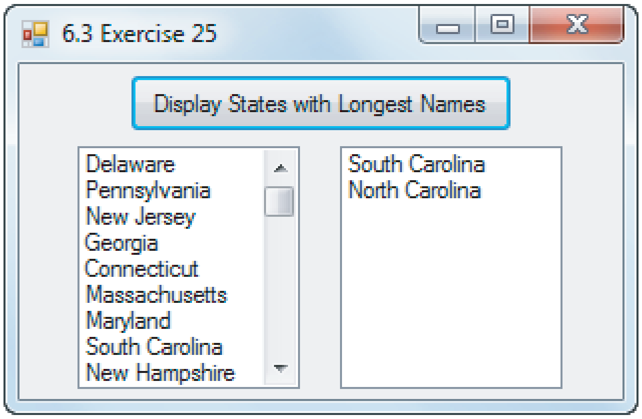
23.
```
Private Sub btnDetermine_Click(...) Handles btnDetermine.Click
  Dim highestIndex As Integer = lstBox.Items.Count - 1
  Dim state As String
  For i As Integer = 0 To highestIndex
    state = CStr(lstBox.Items(i))
    If state.StartsWith("New") Then
      txtOutput.Text = state
      Exit For
    End If
  Next
End Sub
```



Delaware
Pennsylvania
New Jersey
Georgia
Connecticut
Massachusetts
Maryland
South Carolina
New Hampshire

New Jersey

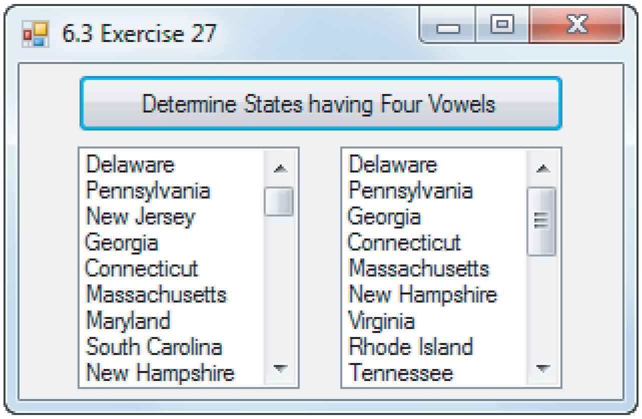
25.
```
Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  Dim highestIndex As Integer = lstBox.Items.Count - 1
  Dim maxLength As Integer = 0
  Dim state As String
  For i As Integer = 0 To highestIndex
    state = CStr(lstBox.Items(i))
    If state.Length > maxLength Then
      maxLength = state.Length
    End If
  Next
  For i As Integer = 0 To highestIndex
    state = CStr(lstBox.Items(i))
    If state.Length = maxLength Then
      lstBox2.Items.Add(state)
    End If
  Next
End Sub
```



Delaware
Pennsylvania
New Jersey
Georgia
Connecticut
Massachusetts
Maryland
South Carolina
New Hampshire

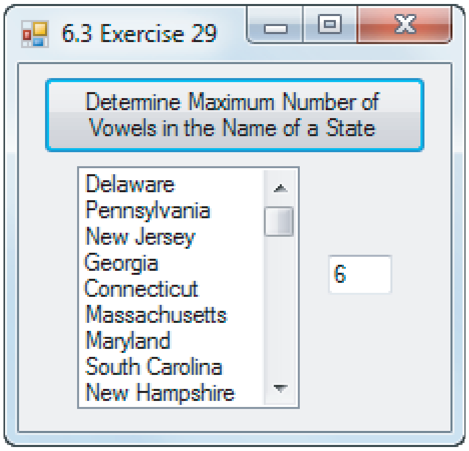
South Carolina
North Carolina

27.
```
Private Sub btnDetermine_Click(...) Handles btnDetermine.Click
  Dim highestIndex As Integer = lstBox.Items.Count - 1
  Dim state As String
  For i As Integer = 0 To highestIndex
    state = CStr(lstBox.Items(i))
    If NumberOfVowels(state) = 4 Then
      lstBox2.Items.Add(state)
    End If
  Next
End Sub

Function NumberOfVowels(ByVal word As String) As Integer
  Dim numVowels As Integer = 0
  word = word.ToUpper
  Dim letter As String
  Dim numLetters As Integer = word.Length
  For i As Integer = 0 To (numLetters - 1)
    letter = word.Substring(i, 1)
    If (letter = "A") Or (letter = "E") Or (letter = "I") Or
       (letter = "O") Or (letter = "U") Then
      numVowels += 1
    End If
  Next
  Return numVowels
End Function
```



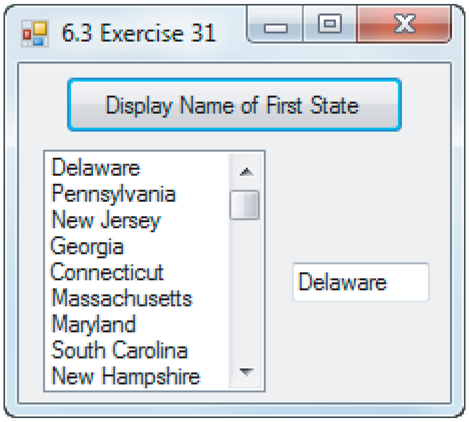
```
Delaware          Delaware
Pennsylvania      Pennsylvania
New Jersey        Georgia
Georgia           Connecticut
Connecticut       Massachusetts
Massachusetts     New Hampshire
Maryland          Virginia
South Carolina    Rhode Island
New Hampshire     Tennessee
```

29.
```
Private Sub btnDetermine_Click(...) Handles btnDetermine.Click
  Dim highestIndex As Integer = lstBox.Items.Count - 1
  Dim state As String
  Dim maxNumOfVowels = 0
  For i As Integer = 0 To highestIndex
    state = CStr(lstBox.Items(i))
    If NumberOfVowels(state) > maxNumOfVowels Then
      maxNumOfVowels = NumberOfVowels(state)
    End If
  Next
  txtOutput.Text = CStr(maxNumOfVowels)
End Sub

Function NumberOfVowels(ByVal word As String) As Integer
  Dim numVowels As Integer = 0
  word = word.ToUpper
  Dim letter As String
  Dim numLetters As Integer = word.Length
  For i As Integer = 0 To (numLetters - 1)
    letter = word.Substring(i, 1)
    If (letter = "A") Or (letter = "E") Or (letter = "I") Or
       (letter = "O") Or (letter = "U") Then
      numVowels += 1
    End If
  Next
  Return numVowels
End Function
```



Delaware
Pennsylvania
New Jersey
Georgia
Connecticut
Massachusetts
Maryland
South Carolina
New Hampshire

6

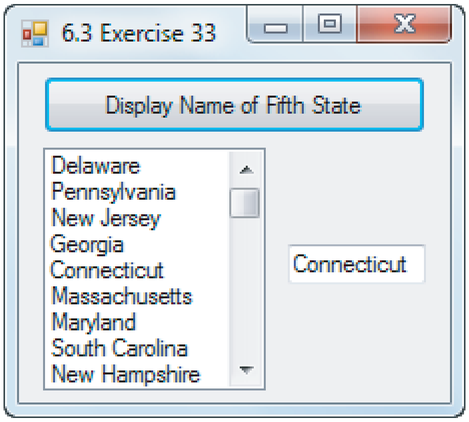


Delaware
Pennsylvania
New Jersey
Georgia
Connecticut
Massachusetts
Maryland
South Carolina
New Hampshire

Delaware

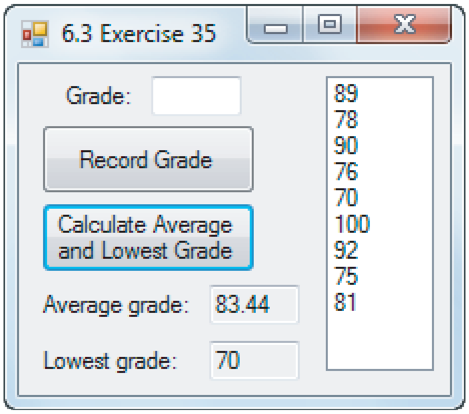
31.
```
Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  txtOutput.Text = CStr(lstBox.Items(0))
End Sub
```

33.
```
Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  txtOutput.Text = CStr(lstBox.Items(4))
End Sub
```



Delaware
Pennsylvania
New Jersey
Georgia
Connecticut
Massachusetts
Maryland
South Carolina
New Hampshire

Connecticut



Grade:

Average grade: 83.44

Lowest grade: 70

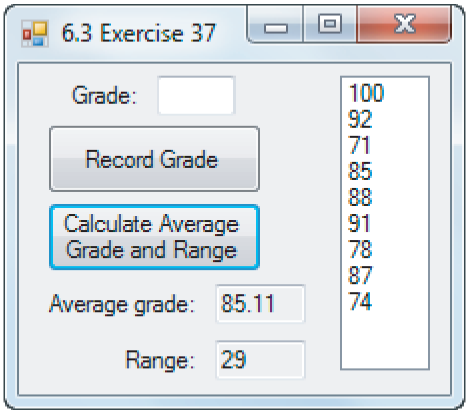
89
78
90
76
70
100
92
75
81

35.
```
Private Sub btnRecord_Click(...) Handles btnRecord.Click
  lstGrades.Items.Add(txtGrade.Text)
  txtGrade.Clear()
  txtGrade.Focus()
End Sub
```

```
Private Sub btnCalculate_Click(...) Handles btnCalculate.Click
  Dim sum As Double = 0
  Dim minGrade As Double = 100
  If lstGrades.Items.Count > 0 Then
    For i As Integer = 0 To lstGrades.Items.Count - 1
      sum += CDbl(lstGrades.Items(i))
      If CDbl(lstGrades.Items(i)) < minGrade Then
        minGrade = CDbl(lstGrades.Items(i))
      End If
    Next
  Else
    MessageBox.Show("You must first enter some grades.")
  End If
  txtAverage.Text = FormatNumber(sum / lstGrades.Items.Count, 2)
  txtLowest.Text = CStr(minGrade)
End Sub
```

37.

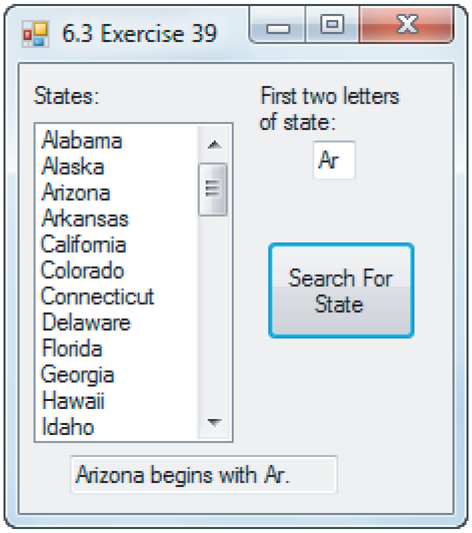
```
Private Sub btnRecord_Click(...) Handles btnRecord.Click
  lstGrades.Items.Add(txtGrade.Text)
  txtGrade.Clear()
  txtGrade.Focus()
End Sub

Private Sub btnCalculate_Click(...) Handles btnCalculate.Click
  Dim sum As Double = 0
  Dim maxGrade As Double = 0
  Dim minGrade As Double = 100
  If lstGrades.Items.Count > 0 Then
    For i As Integer = 0 To lstGrades.Items.Count - 1
      sum += CDbl(lstGrades.Items(i))
      If CDbl(lstGrades.Items(i)) > maxGrade Then
        maxGrade = CDbl(lstGrades.Items(i))
      End If
      If CDbl(lstGrades.Items(i)) < minGrade Then
        minGrade = CDbl(lstGrades.Items(i))
      End If
    Next
  Else
    MessageBox.Show("You must first enter some grades.")
  End If
  txtAverage.Text = FormatNumber(sum / lstGrades.Items.Count, 2)
  txtRange.Text = CStr(maxGrade - minGrade)
End Sub
```



Average grade: 85.11

Range: 29

```
100
92
71
85
88
91
78
87
74
```



```
Alabama
Alaska
Arizona
Arkansas
California
Colorado
Connecticut
Delaware
Florida
Georgia
Hawaii
Idaho
```

First two letters of state: Ar

Arizona begins with Ar.

39.
```
Private Sub btnSearch_Click(...) Handles btnSearch.Click
  Dim letters As String = mtbFirstTwoLetters.Text.ToUpper
  Dim i As Integer = 49     'index of the state currently considered
  Do Until (CStr(lstStates.Items(i)).ToUpper <= letters) Or (i = 0)
    i = i – 1
  Loop
  If CStr(lstStates.Items(i + 1)).ToUpper.StartsWith(letters) Then
    txtOutput.Text = CStr(lstStates.Items(i + 1)) & " begins with " &
                     mtbFirstTwoLetters.Text & "."
  ElseIf CStr(lstStates.Items(0)).ToUpper.StartsWith(letters) Then
    txtOutput.Text = CStr(lstStates.Items(0)) & " begins with " &
                     mtbFirstTwoLetters.Text & "."
  Else
    txtOutput.Text = "No state begins with " &
                      mtbFirstTwoLetters.Text & "."
  End If
End Sub
```

Chapter 7

Exercises 7.1

1. `101`   3. `Have a dessert spoon.`   5. `Yes`   7. `12`

9. `You have a trio.`   11. `Your average is 80`

13. `Slumdog Millionaire won in 2009`   15. `one,two,three`

17. `2 even numbers`   19. `Pearl Harbor: 1941`

21. `contains a 19th-century date`   23. `6 words begin with a vowel`

25.
```
4
6
2
```

27. a. `Superior` (last name in alphabetical order)
b. `Erie` (first name in alphabetical order)
c. `Huron` (first name in the array)
d. `Superior` (last name in the array)
e. `5` (number of names in the array)
f. `Ontario` (second name in the array)
g. `3` (first array subscript whose element is Erie)

29. a. `6.5` (greatest population of a New England state)
b. `0.7` (least population of a New England state)
c. `3.5` (first population in the array)
d. `1.3` (last population in the array)
e. `6` (number of numbers in the array)
f. `1.1` (fourth population in the array)
g. `3` (first array subscript whose element is 1.1)

31. a. `lstOutput.Items.Add(states.First)`
or `lstOutput.Items.Add(states(0))`

b.
```
For i As Integer = 0 To 12
  lstOutput.Items.Add(states(i))
Next
```