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
```

c. `lstOutput.Items.Add(states.Last)`
   or `lstOutput.Items.Add(states(49))`

d. `lstOutput.Items.Add(CStr(Array.IndexOf(states, "Ohio") + 1))`

e. `lstOutput.Items.Add(states(1))`

f. `lstOutput.Items.Add(states(19))`

g.
```
For i As Integer = (states.Count − 9) To (states.Count)
  lstOutput.Items.Add(states(i − 1))
Next
```

33.
```
Function Task(ByVal nums() As Integer) As Integer
  Dim sum As Integer = 0
  For Each num As Integer In nums
    sum += num
  Next
  Return sum
End Function
```

35.
```
Function Task(ByVal nums() As Integer) As Integer
  Dim maxEven As Integer = 0
  For Each num As Integer In nums
    If (num Mod 2 = 0) And (num > maxEven) Then
      maxEven = num
    End If
  Next
  Return maxEven
End Function
```

37.
```
Function Task(ByVal nums() As Integer) As Integer
  Dim twoDigits As Integer = 0
  For Each num As Integer In nums
    If (num > 9) And (num < 100) Then
      twoDigits += 1
    End If
  Next
  Return twoDigits
End Function
```

39. `nums(3)` should be changed to `nums()`

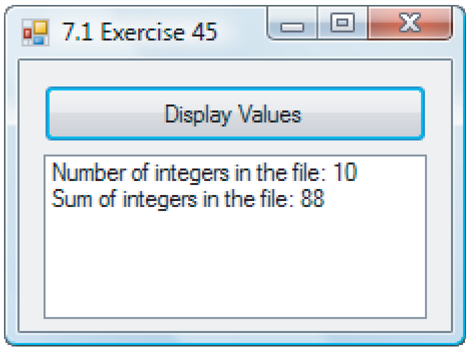
41. Logic error. The values of the array elements cannot be altered inside a For Each loop. The output will be 6.

43. `lstBox.Items.Add(line.Split(" "c).Count)`

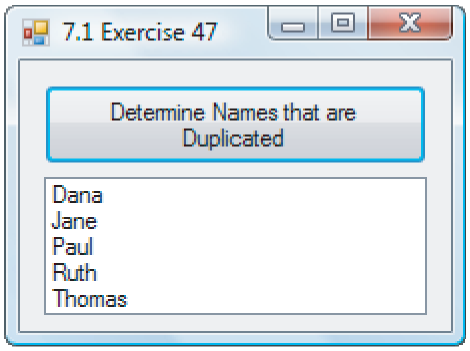
45.
```
Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  Dim numStr() As String = IO.File.ReadAllLines("Numbers.txt")
  Dim nums(numStr.Count − 1) As Integer
  For i As Integer = 1 To nums.Count − 1
    nums(i) = CInt(numStr(i))
  Next
  lstOutput.Items.Add("Number of integers in the file: " & nums.Count)
  lstOutput.Items.Add("Sum of integers in the file: " & nums.Sum)
End Sub
```

47.
```
Private Sub btnDetermine_Click(...) Handles btnDetermine.Click
  Dim names() As String = IO.File.ReadAllLines("Names2.txt")
  Dim dups(names.Count − 1) As String
  Dim n As Integer = 0         'index for dups
  For i As Integer = 0 To names.Count − 2
    If (names(i + 1) = names(i)) And
                     (Array.IndexOf(dups, names(i)) = −1) Then
```

```
        dups(n) = names(i)
        n += 1
      End If
    Next
    If n = 0 Then
      lstOutput.Items.Add("No duplicates.")
    Else
      For i As Integer = 0 To n - 1
        lstOutput.Items.Add(dups(i))
      Next
    End If
  End Sub
```

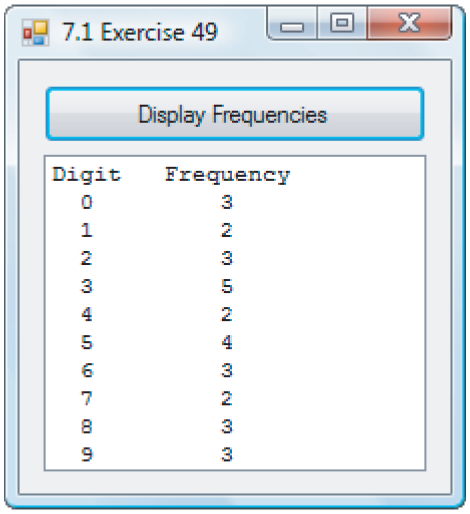


```
Number of integers in the file: 10
Sum of integers in the file: 88
```



```
Dana
Jane
Paul
Ruth
Thomas
```

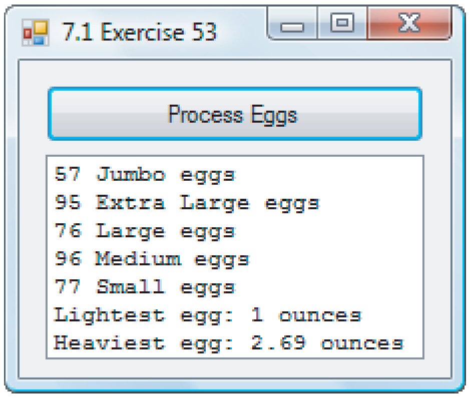
49.
```
Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  Dim strDigits() As String = IO.File.ReadAllLines("Digits.txt")
  Dim freq(9) As Integer
  For i As Integer = 0 To strDigits.Count - 1
    freq(CInt(strDigits(i))) += 1
  Next
  lstOutput.Items.Add("Digit   Frequency")
  For i As Integer = 0 To 9
    lstOutput.Items.Add("  " & i & "         " & freq(i))
  Next
End Sub
```



```
Digit   Frequency
  0         3
  1         2
  2         3
  3         5
  4         2
  5         4
  6         3
  7         2
  8         3
  9         3
```

51.
```
Function Sum(ByVal nums() As Integer) As Integer
  Dim total As Integer = 0
  For i As Integer = 1 To nums.Count - 1 Step 2
    total += nums(i)
  Next
  Return total
End Function
```

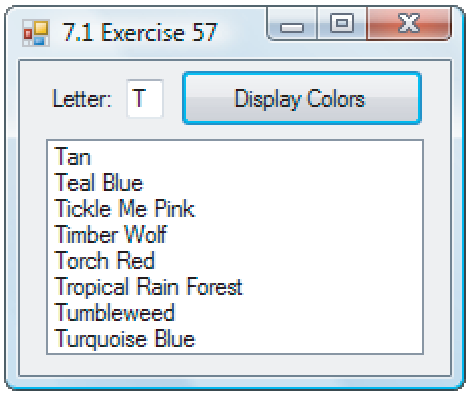
53.
```
Private Sub btnProcessEggs_Click(...) Handles btnProcessEggs.Click
  Dim heaviest, lightest, ounces As Double
  Dim jumbo, xLarge, large, med, small As Integer
  heaviest = 0     'can be any number lower than lightest egg
  lightest = 100  'can be any number greater than heaviest egg
  Dim strEggs() As String = IO.File.ReadAllLines("Eggs.txt")
  Dim eggs(strEggs.Count - 1) As Double
  For i As Integer = 0 To eggs.Count - 1
    eggs(i) = CDbl(strEggs(i))
  Next
  For i As Integer = 0 To eggs.Count - 1
    ounces = eggs(i)
    If ounces > heaviest Then
      heaviest = ounces
    End if
    If ounces < lightest Then
      lightest = ounces
    End If
    Select Case ounces
      Case Is < 1.5
        'too small & cannot be sold
      Case Is < 1.75
        small += 1
      Case Is < 2
        med += 1
      Case Is < 2.25
        large += 1
      Case Is < 2.5
        xLarge += 1
      Case Else
        jumbo += 1
    End Select
  Next
  lstOutput.Items.Clear()
  lstOutput.Items.Add(jumbo & " Jumbo eggs")
  lstOutput.Items.Add(xLarge & " Extra Large eggs")
  lstOutput.Items.Add(large & " Large eggs")
  lstOutput.Items.Add(med & " Medium eggs")
  lstOutput.Items.Add(small & " Small eggs")
  If lightest <> 100 Then
    lstOutput.Items.Add("Lightest egg: " & lightest & " ounces")
    lstOutput.Items.Add("Heaviest egg: " & heaviest & " ounces")
  Else
    lstOutput.Items.Add("File is empty")
  End If
End Sub
```



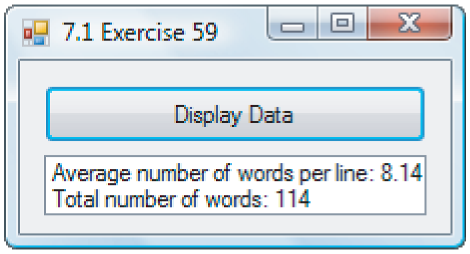
```
57 Jumbo eggs
95 Extra Large eggs
76 Large eggs
96 Medium eggs
77 Small eggs
Lightest egg: 1 ounces
Heaviest egg: 2.69 ounces
```

55.
```
Dim colors() As String = IO.File.ReadAllLines("Colors.txt")

Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  Dim letter As String = mtbLetter.Text.ToUpper
  lstColors.Items.Clear()
  For Each hue As String In colors
    If hue.StartsWith(letter) Then
      lstColors.Items.Add(hue)
    End If
  Next
End Sub
```

57.
```
Dim colors() As String = IO.File.ReadAllLines("Colors.txt")

Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  Dim letter As String = mtbLetter.Text.ToUpper     'mask L
  lstColors.Items.Clear()
  For Each hue As String In SmallerArray(letter)
    lstColors.Items.Add(hue)
  Next
End Sub

Function SmallerArray(ByVal letter As String) As String()
  Dim smArray(colors.Count - 1) As String
  Dim counter As Integer = 0
  For Each hue As String In colors
    If hue.StartsWith(letter) Then
      smArray(counter) = hue
      counter += 1
    End If
  Next
  ReDim Preserve smArray(counter - 1)
  Return smArray
End Function
```



```
Tan
Teal Blue
Tickle Me Pink
Timber Wolf
Torch Red
Tropical Rain Forest
Tumbleweed
Turquoise Blue
```



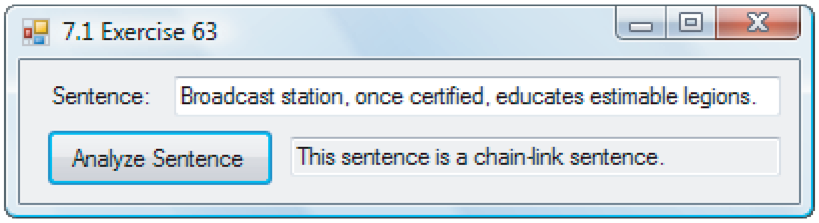
```
Average number of words per line: 8.14
Total number of words: 114
```

59.
```
Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  Dim lines() As String = IO.File.ReadAllLines("Sonnet.txt")
  Dim n = lines.Count - 1
  Dim numWords(n) As Integer
  For i As Integer = 0 To n
    numWords(i) = lines(i).Split(" "c).Count
  Next
  lstOutput.Items.Add("Average number of words per line: " &
                      FormatNumber(numWords.Average, 2))
  lstOutput.Items.Add("Total number of words: " & numWords.Sum)
End Sub
```

61.
```
Dim grades(99) As Integer       'stores grades
Dim numGrades As Integer        'number of grades stored

Private Sub btnRecord_Click(...) Handles btnRecord.Click
  'Add a score to the array
  'If no more room, then display error message.
  If numGrades >= 100 Then
    MessageBox.Show("100 scores have been entered.", "No more room.")
  Else
    grades(numGrades) = CInt(txtScore.Text)
    numGrades += 1
    lstOutput.Items.Clear()
    txtScore.Clear()
    txtScore.Focus()
  End If
End Sub

Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  'Display average of grades and the number of above average grades
  Dim temp() As Integer = grades
  ReDim Preserve temp(numGrades – 1)
  lstOutput.Items.Clear()
  lstOutput.Items.Add("The average grade is " &
                      FormatNumber(temp.Average, 2) & ".")
  lstOutput.Items.Add(NumAboveAverage(temp) &
                     " students scored above the average.")
End Sub

Function NumAboveAverage(ByVal temp() As Integer) As Integer
  'Count the number of scores above the average grade
  Dim avg As Double = temp.Average
  Dim num As Integer = 0
  For Each grade In temp
    If grade > avg Then
      num += 1
    End If
  Next
  Return num
End Function
```

63.
```
Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  If IsChainLink(txtSentence.Text) Then
    txtOutput.Text = "This sentence is a chain-link sentence."
  Else
    txtOutput.Text = "This sentence is not a chain-link sentence."
  End If
End Sub

Function IsChainLink(ByVal sentence As String) As Boolean
  'Analyze a sentence to see whether it is a chain-link sentence.
  Dim words(), ending As String
  'Split the sentence into words, removing commas first
  words = txtSentence.Text.Replace(",", "").Split(" "c)
  For i As Integer = 0 To words.Count – 2
    If (words(i).Length < 2) Or (words(i + 1).Length < 2) Then
      Return False    'If any word has is less than two letters.
    End If
    ending = words(i).Substring(words(i).Length – 2).ToUpper
```

```
      If ending <> words(i + 1).Substring(0, 2).ToUpper Then
        Return False    'If ending does not match beginning of next word.
      End If
    Next
    Return True  'If all words are ok, then it is a chain-link sentence.
  End Function
```



Broadcast station, once certified, educates estimable legions.
This sentence is a chain-link sentence.

Exercises 7.2

1. 5
7

3. going
offer
can't

5. 6

7. 103

9. 8

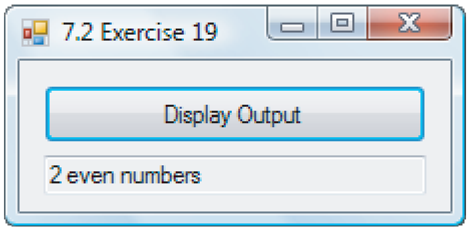
11. 3 students have a grade of 100

13. 15
12

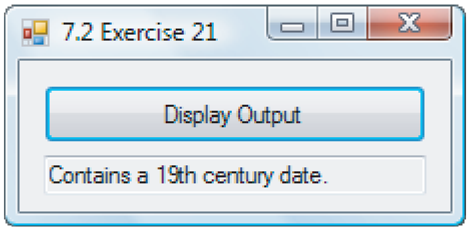
15. The average after dropping the lowest grade is 80

17. 37 is a prime number

19.
```
Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  Dim nums() As Integer = {3, 5, 8, 10, 21}
  Dim query = From num In nums
              Where num Mod 2 = 0
              Select num
  txtOutput.Text = query.count & " even numbers"
End Sub
```



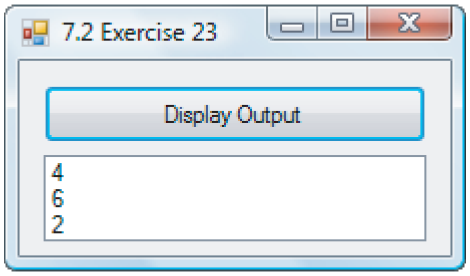
2 even numbers



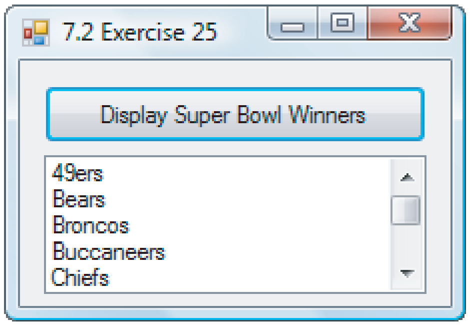
Contains a 19th century date.

21.
```
Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  Dim dates() As String = IO.File.ReadAllLines("Dates.txt")
  Dim query = From yr In dates
              Where (CInt(yr) >= 1800) And (CInt(yr) <= 1899)
              Select yr
  If query.Count > 0 Then
    txtOutput.Text = "contains a 19th century date."
  Else
    txtOutput.Text = "does not contain a 19th century date."
  End If
End Sub
```

23.
```
Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  Dim nums() As Integer = {2, 6, 4}
  Dim query = From num In nums
              Order By Array.IndexOf(nums, num) Descending
  For Each num As Integer In query
    lstOutput.Items.Add(num)
  Next
End Sub
```



```
4
6
2
```



```
49ers
Bears
Broncos
Buccaneers
Chiefs
```

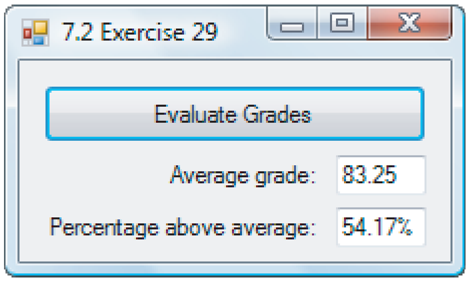
25.
```
Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  Dim teams() As String = IO.File.ReadAllLines("SBWinners.txt")
  Dim query = From team In teams
              Order By team Ascending
              Distinct
  For Each team As String In query
    lstOutput.Items.Add(team)
  Next
End Sub
```

27.
```
Dim teamNames() As String = IO.File.ReadAllLines("SBWinners.txt")

Private Sub btnDetermine_Click(...) Handles btnDetermine.Click
  'Display the number of Super Bowls won by the team in the text box
  Dim query = From team In teamNames
              Where team.ToUpper = txtName.Text.ToUpper
              Select team
  txtNumWon.Text = CStr(query.Count)
End Sub
```

29.
```
Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  Dim query1 = From grade In IO.File.ReadAllLines("Final.txt")
               Select CInt(grade)
  Dim avg As Double = query1.Average
  Dim query2 = From grade In IO.File.ReadAllLines("Final.txt")
               Where CInt(grade) > avg
               Select grade
  txtAverage.Text = FormatNumber(avg)
  txtAboveAve.Text = FormatPercent(query2.Count / query1.Count)
End Sub
```

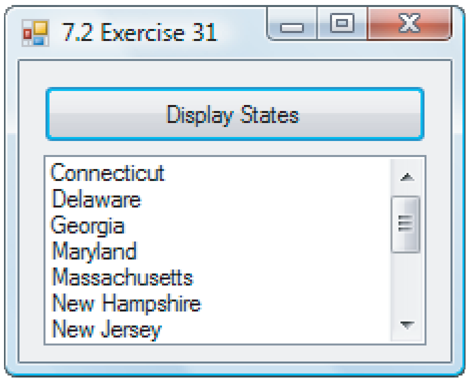
31.
```
Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  Dim states() As String = IO.File.ReadAllLines("States.txt")
  ReDim Preserve states(12)
  Dim query = From state In states
              Order By state
              Select state
```

```
    For Each state As String In query
      lstOutput.Items.Add(state)
    Next
  End Sub
```



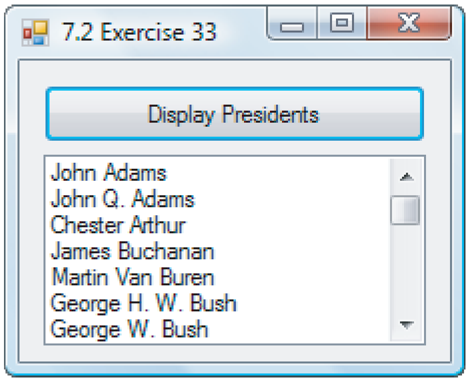
Average grade: 83.25

Percentage above average: 54.17%



```
Connecticut
Delaware
Georgia
Maryland
Massachusetts
New Hampshire
New Jersey
```

33.
```
  Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
    Dim query = From pres In IO.File.ReadAllLines("USPres.txt")
                Let lastName = pres.Split(" "c).Last
                Order By lastName
                Select pres
    For Each pres As String In query
      lstOutput.Items.Add(pres)
    Next
    Distinct
  End Sub
```



```
John Adams
John Q. Adams
Chester Arthur
James Buchanan
Martin Van Buren
George H. W. Bush
George W. Bush
```

35.
```
  Dim nations() As String = IO.File.ReadAllLines("Nations.txt")

  Private Sub frmNations_Load(...) Handles MyBase.Load
    lstNations.DataSource = nations
    lstNations.SelectedItem = Nothing
  End Sub

  Private Sub txtNations_TextChanged(...) Handles txtNation.TextChanged
    Dim query = From nation In nations
                Where nation.StartsWith(txtNation.Text)
                Select nation
    lstNations.DataSource = query.ToList
    lstNations.SelectedItem = Nothing
  End Sub
```

```
Private Sub lstNations_Click(...) Handles lstNations.Click
  txtNation.Text = lstNations.Text
End Sub
```

Exercises 7.3

1. `The area of a football field is 19200 square yards.`

3. `Duke was founded in NC in 1838.`

5.
```
heights are same
 170
```

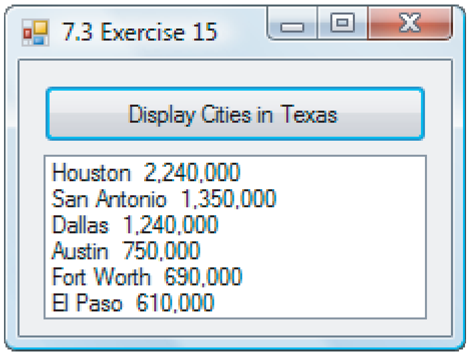
7.
```
Joe: 88
Moe: 90
Roe: 95
```

9. `Mr. President lives in Washington, DC`

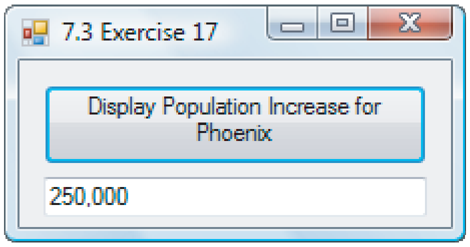
11. In the event procedure, `peace` should be `prize.peace` and `yr` should be `prize.yr`.

13. The condition `(game1 > game2)` is not valid. Structures can only be compared one field at a time.

15. The cities in Texas, along with their populations. The cities are ordered by the sizes of their populations beginning with the most populous city.



```
Houston  2,240,000
San Antonio  1,350,000
Dallas  1,240,000
Austin  750,000
Fort Worth  690,000
El Paso  610,000
```



```
250,000
```

17. The population growth of Phoenix from 2000 to 2010.

19.
```
Structure State
  Dim name As String
  Dim abbreviation As String
  Dim area As Double
  Dim pop As Double
End Structure

Dim states() As State

Private Sub frmStates_Load(...) Handles MyBase.Load
  Dim stateRecords() As String = IO.File.ReadAllLines("USStates.txt")
  Dim n As Integer = stateRecords.Count – 1
  ReDim states(n)
  Dim line As String
  Dim data() As String
  For i As Integer = 0 To n
    line = stateRecords(i)
    data = line.Split(","c)
    states(i).name = data(0)
    states(i).abbreviation = data(1)
    states(i).area = CDbl(data(2))
    states(i).pop = CDbl(data(3))
  Next
End Sub

Private Sub btnFind_Click(...) Handles btnFind.Click
  Dim stateAbbr As String = mtbAbbrev.Text.ToUpper
  Dim query = From state In states
              Where state.abbreviation = stateAbbr
              Select state.name, state.area
  txtOutput.Text = "The area of " & query.First.name & " is " &
                    FormatNumber(query.First.area, 0) & " square miles."
End Sub
```

21. (Begin with the code from Exercise 19 and replace the Click event procedure with the following.)

```
Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  Dim query = From state In states
              Let density = state.pop / state.area
              Let formattedDensity = FormatNumber(density, 2)
              Order By density Descending
              Select state.name, formattedDensity
  dgvOutput.DataSource = query.ToList
  dgvOutput.CurrentCell = Nothing
  dgvOutput.Columns("name").HeaderText = "State"
  dgvOutput.Columns("formattedDensity").HeaderText =
                        "People per Square Mile"
End Sub
```

23.

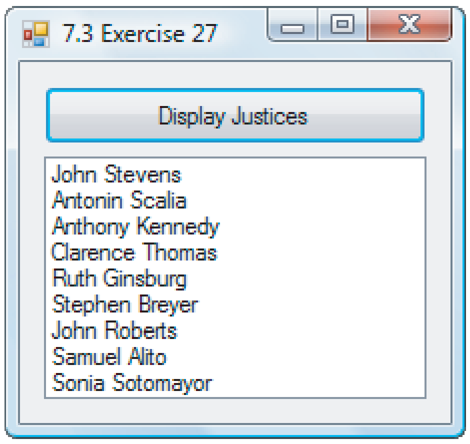
```
Structure Player
  Dim name As String
  Dim team As String
  Dim atBats As Double
  Dim hits As Double
End Structure

Dim players() As Player

Private Sub frmBaseball_Load(...) Handles MyBase.Load
  Dim playerStats() As String = IO.File.ReadAllLines("Baseball.txt")
  Dim n As Integer = playerStats.Count – 1
  ReDim players(n)
  Dim line As String
  Dim data() As String
  For i As Integer = 0 To n
    line = playerStats(i)
    data = line.Split(","c)
    players(i).name = data(0)
    players(i).team = data(1)
    players(i).atBats = CDbl(data(2))
    players(i).hits = CDbl(data(3))
  Next
  Dim query = From person In players
              Order By person.team Ascending
              Select person.team
              Distinct
  lstTeams.DataSource = query.ToList
End Sub
```

```
    Private Sub lstTeams_SelectedIndexChanged(...) Handles _
                             lstTeams.SelectedIndexChanged
      Dim selectedTeam = lstTeams.Text
      Dim query = From person In players
                  Where person.team = selectedTeam
                  Order By person.hits Descending
                  Select person.name, person.hits
      dgvOutput.DataSource = query.ToList
      dgvOutput.CurrentCell = Nothing
      dgvOutput.Columns("name").HeaderText = "Player"
      dgvOutput.Columns("hits").HeaderText = "Hits"
    End Sub
```

25.
```
Structure Player
  Dim name As String
  Dim team As String
  Dim atBats As Double
  Dim hits As Double
End Structure

Dim players() As Player

Private Sub frmBaseball_Load(...) Handles MyBase.Load
  Dim playerStats() As String = IO.File.ReadAllLines("Baseball.txt")
  Dim n As Integer = playerStats.Count – 1
  ReDim players(n)
  Dim line As String
  Dim data() As String
  For i As Integer = 0 To n
    line = playerStats(i)
    data = line.Split(","c)
    players(i).name = data(0)
    players(i).team = data(1)
    players(i).atBats = CDbl(data(2))
    players(i).hits = CDbl(data(3))
  Next
End Sub

Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  Dim query = From person In players
              Let ave = person.hits / person.atBats
              Select ave
  Dim best As Double = query.Max
  txtBestAverage.Text = FormatNumber(best, 3)
  Dim query2 = From person In players
               Where person.hits / person.atBats = best
               Select person.name, person.team
  dgvOutput.DataSource = query2.ToList
  dgvOutput.CurrentCell = Nothing
  dgvOutput.Columns("name").HeaderText = "Player"
  dgvOutput.Columns("team").HeaderText = "Team"
End Sub
```

27.
```
Structure Justice
  Dim firstName As String
  Dim lastName As String
  Dim apptPres As String
  Dim state As String     'state abbreviation
  Dim yrAppointed As Double
  Dim yrLeft As Double
End Structure
```

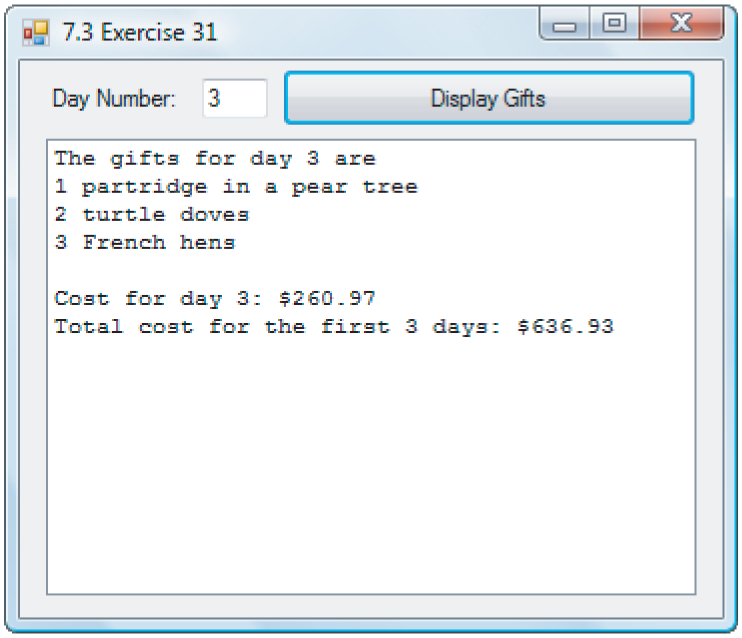
```
Dim justices() As Justice

Private Sub frmJustices_Load(...) Handles MyBase.Load
  Dim justiceRecords() As String = IO.File.ReadAllLines("Justices.txt")
  Dim n As Integer = justiceRecords.Count - 1
  ReDim justices(n)
  Dim line As String
  Dim data() As String
  For i As Integer = 0 To n
    line = justiceRecords(i)
    data = line.Split(","c)
    justices(i).firstName = data(0)
    justices(i).lastName = data(1)
    justices(i).apptPres = data(2)
    justices(i).state = data(3)
    justices(i).yrAppointed = CDbl(data(4))
    justices(i).yrLeft = CDbl(data(5))
  Next
End Sub

Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  Dim query = From person In justices
              Where person.yrLeft = 0
              Order By person.yrAppointed
              Select person.firstName & " " & person.lastName
  lstOutput.DataSource = query.ToList
  lstOutput.SelectedItem = Nothing
End Sub
```



```
John Stevens
Antonin Scalia
Anthony Kennedy
Clarence Thomas
Ruth Ginsburg
Stephen Breyer
John Roberts
Samuel Alito
Sonia Sotomayor
```

29. (Begin with the code from Exercise 27 and replace the Click event procedure with the following.)

```
Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  Dim query = From person In justices
              Where person.state = mtbState.Text
              Let fullName = person.firstName & " " & person.lastName
              Let yrs = YearsServed(person.yrAppointed, person.yrLeft)
              Let presLastName = person.apptPres.Split(" "c).Last
              Select fullName, presLastName, yrs
  If query.Count = 0 Then
    MessageBox.Show("No justices appointed from that state.", "NONE")
    mtbState.Focus()
  Else
    dgvOutput.DataSource = query.ToList
    dgvOutput.CurrentCell = Nothing
    dgvOutput.Columns("fullName").HeaderText = "Justice"
    dgvOutput.Columns("presLastName").HeaderText = "Appointing President"
    dgvOutput.Columns("yrs").HeaderText = "Years Served"
  End If
End Sub

Function YearsServed(ByVal enter As Double,
                     ByVal leave As Double) As Double
  If leave = 0 Then
    Return (Now.Year – enter)
  Else
    Return (leave – enter)
  End If
End Function
```



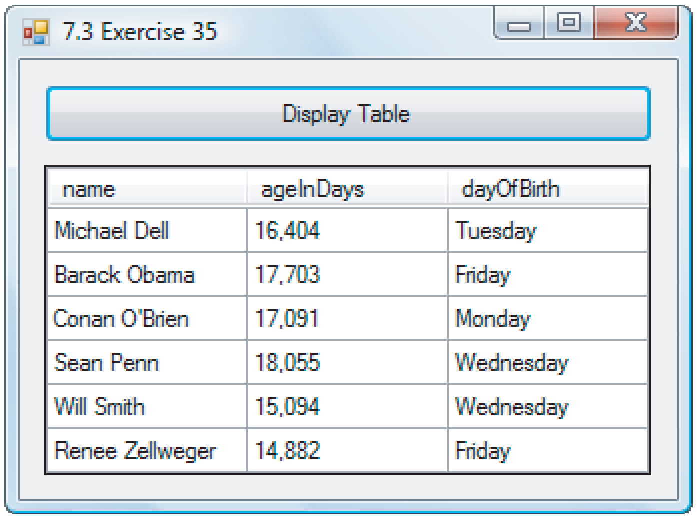
```
The gifts for day 3 are
1 partridge in a pear tree
2 turtle doves
3 French hens

Cost for day 3: $260.97
Total cost for the first 3 days: $636.93
```

31.
```
Structure Day
  Dim num As Integer
  Dim present As String
  Dim price As Double
End Structure

Dim days() As Day

Private Sub frmXmas_Load(...) Handles MyBase.Load
  Dim gifts() As String = IO.File.ReadAllLines("Gifts.txt")
  Dim n As Integer = gifts.Count – 1
  ReDim days(n)
  Dim data() As String
  For i As Integer = 0 To n
    data = gifts(i).Split(","c)
    days(i).num = CInt(data(0))
    days(i).present = data(1)
    days(i).price = CDbl(data(2))
  Next
End Sub
```

```
Private Sub btnDisplayGifts_Click(...) Handles btnDisplayGifts.Click
  Dim dayNum = CInt(txtDayNum.Text)
  Dim cost As Double = 0
  Dim totalCost As Double = 0
  lstOutput.Items.Clear()
  lstOutput.Items.Add("The gifts for day " & dayNum & " are")
  For i As Integer = 0 To (dayNum - 1)
    lstOutput.Items.Add(days(i).num & " " & days(i).present)
    cost += days(i).num * days(i).price
    totalCost += days(i).num * days(i).price *
                 (dayNum + 1 - days(i).num)
  Next
  lstOutput.Items.Add("")
  lstOutput.Items.Add("Cost for day " & dayNum & ": " &
                      FormatCurrency(cost))
  lstOutput.Items.Add("Total cost for the first " & dayNum &
                      " days: " & FormatCurrency(totalCost))
End Sub
```

33.
```
Structure FamousPerson
  Dim name As String
  Dim dateOfBirth As Date
End Structure

Dim famousPersons() As FamousPerson

Private Sub frmFamous_Load(...) Handles MyBase.Load
  Dim people() As String = IO.File.ReadAllLines("Famous.txt")
  Dim n As Integer = people.Count - 1
  ReDim famousPersons(n)
  Dim line As String
  Dim data() As String
  For i As Integer = 0 To n
    line = people(i)
    data = line.Split(","c)
    famousPersons(i).name = data(0)
    famousPersons(i).dateOfBirth = CDate(data(1))
  Next
End Sub

Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  Dim query = From person In famousPersons
              Where (person.dateOfBirth >= #1/1/1970#) And
                    (person.dateOfBirth < #1/1/1980#)
              Select person.name
  lstOutput.DataSource = query.ToList
  lstOutput.SelectedItem = Nothing
End Sub
```

35.
```
Dim people() As Person

Private Sub frmFamous_Load(...) Handles MyBase.Load
  'Place the data for each person into the array people.
  Dim group() As String = IO.File.ReadAllLines("Famous.txt")
  Dim n As Integer = group.Count - 1
  ReDim people(n)
  Dim data() As String
  For i As Integer = 0 To n
    data = group(i).Split(","c)
    people(i).name = data(0)
    people(i).dateOfBirth = CDate(data(1))
  Next
End Sub

Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  Dim query = From individual In people
              Let ageInDays = FormatNumber(DateDiff(DateInterval.Day,
                  individual.dateOfBirth, Today), 0)
              Let dayOfBirth = DayOfWeek(individual.dateOfBirth)
              Where individual.dateOfBirth.AddYears(40) <= Today And
                    individual.dateOfBirth.AddYears(50) > Today
              Select individual.name, ageInDays, dayOfBirth
  dgvOutput.DataSource = query.ToList
  dgvOutput.CurrentCell = Nothing
End Sub

Function DayOfWeek(ByVal d As Date) As String
  Dim d1 As String = FormatDateTime(d, DateFormat.LongDate)
  Dim d2() As String = d1.Split(","c)
  Return First
End Function
```



name	ageInDays	dayOfBirth
Michael Dell	16,404	Tuesday
Barack Obama	17,703	Friday
Conan O'Brien	17,091	Monday
Sean Penn	18,055	Wednesday
Will Smith	15,094	Wednesday
Renee Zellweger	14,882	Friday

37.
```
Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  lstOutput.Items.Clear()
  For i As Integer = 0 To club.Count – 1
    If club(i).courses.Count = 3 Then
      lstOutput.Items.Add(club(i).name)
    End If
  Next
End Sub
```

39.
```
Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  'Displays the students who are not enrolled in CMSC 100
  Dim subject = "CMSC 100"
  Dim ub = club.Count – 1
  Dim checkList(ub) As Boolean
  For i As Integer = 0 To ub
    For j As Integer = 0 To club(i).courses.Count – 1
```

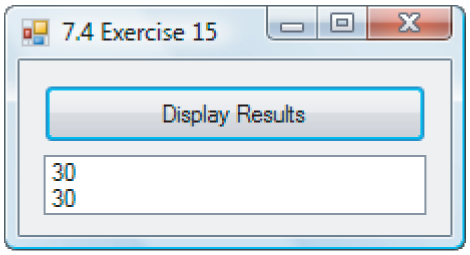
```
            If club(i).courses(j) = subject Then
              checkList(i) = True
            End If
          Next
        Next
        For i As Integer = 0 To ub
          If Not checkList(i) Then
            lstOutput.Items.Add(club(i).name)
          End If
        Next
      End Sub
```

EXERCISES 7.4

1. 1 **3.** 3 **5.** 55 **7.** 14 **9.** 2 **11.** 55

13.
```
Dim twice(2, 3) As Double
For r As Integer = 0 To 2
  For c As Integer = 0 To 3
    twice(r, c) = 2 * nums(r, c)
  Next
Next
```

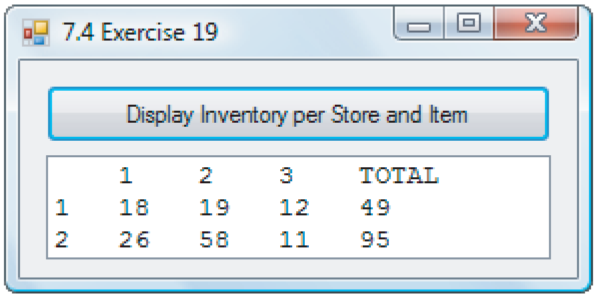
15.
```
'use a For Each loop
Dim total As Double = 0
For Each num As Double In nums
  If num Mod 2 = 0 Then
    total += num
  End If
Next
lstOutput.Items.Add(total)

'use LINQ
Dim query = From num In nums.Cast(Of Double)()
            Where (num Mod 2 = 0)
            Select num
lstOutput.Items.Add(query.Sum)
```



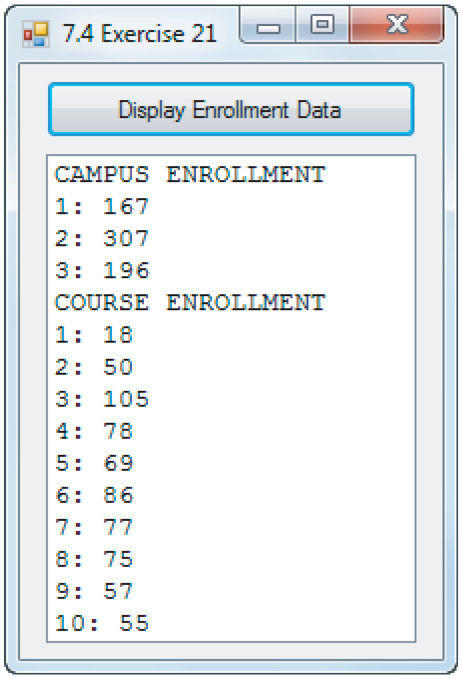
```
30
30
```

17. 12

19.
```
Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  'Display a company's inventory from its two stores
  Dim inventory(,) As Integer = {{25, 64, 23}, {30, 82, 19}}
  Dim sales(,) As Integer = {{7, 45, 11}, {4, 24, 8}}
  Dim total(2) As Integer
  'Adjust the inventory values to reflect today's sales
  For store As Integer = 1 To 2
    For item As Integer = 1 To 3
      inventory(store - 1, item - 1) =
          inventory(store – 1, item – 1) – sales(store – 1, item – 1)
      'Accumulate the total inventory per store
      total(store) += inventory(store – 1, item – 1)
    Next
  Next
  'Display the store's inventory and totals
  lstOutput.Items.Add("    1     2     3     TOTAL")
  For store As Integer = 1 To 2
    lstOutput.Items.Add(store & "    " & inventory(store – 1, 0) &
                      "    " & inventory(store – 1, 1) & "    " &
                      inventory(store – 1, 2) & "    " & total(store))
  Next
End Sub
```



```
     1     2     3     TOTAL
1    18    19    12    49
2    26    58    11    95
```

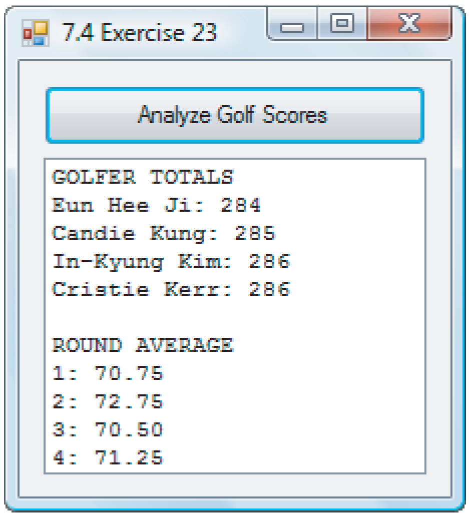


```
CAMPUS ENROLLMENT
1: 167
2: 307
3: 196
COURSE ENROLLMENT
1: 18
2: 50
3: 105
4: 78
5: 69
6: 86
7: 77
8: 75
9: 57
10: 55
```

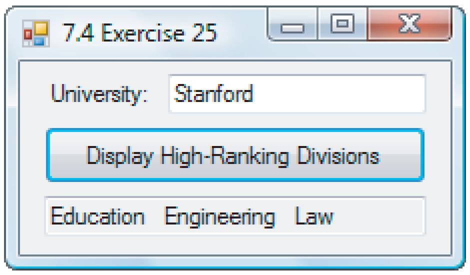
21.
```
Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  'Display the course and campus enrollments
  'enrollment array named er
  Dim er(,) As Integer = {{5, 15, 22, 21, 12, 25, 16, 11, 17, 23},
                          {11, 23, 51, 25, 32, 35, 32, 52, 25, 21},
                          {2, 12, 32, 32, 25, 26, 29, 12, 15, 11}}
  'Define the arrays to accumulate the information
  Dim campusTotal(2), courseTotal(9) As Integer
  For campus As Integer = 0 To 2
    For course As Integer = 0 To 9
      campusTotal(campus) += er(campus, course)
      courseTotal(course) += er(campus, course)
    Next
  Next
  'Display the campus enrollment
  lstOutput.Items.Add("CAMPUS ENROLLMENT")
  For campus As Integer = 0 To 2
    lstOutput.Items.Add((campus + 1) & ": " & campusTotal(campus))
  Next
  'Display the course enrollment
```

```
    lstOutput.Items.Add("COURSE ENROLLMENT")
    For course As Integer = 0 To 9
      lstOutput.Items.Add((course + 1) & ": " & courseTotal(course))
    Next
  End Sub
```

23.
```
  Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
    'Load golf data, cumulate totals, and display results
    Dim scores(3, 3) As Integer
    Dim golfers(3) As String
    Dim table() As String = IO.File.ReadAllLines("Golf.txt")
    Dim data() As String
    Dim golferTotal(3) As Integer, roundTotal(3) As Integer
    For i As Integer = 0 To 3
      data = table(i).Split(","c)
      golfers(i) = data(0)
      For j = 0 To 3
        scores(i, j) = CInt(data(j + 1))
      Next
    Next
    For golfer As Integer = 0 To 3
      For round As Integer = 0 To 3
        golferTotal(golfer) += scores(golfer, round)
        roundTotal(round) += scores(golfer, round)
      Next
    Next
    'Display golfer's totals
    lstOutput.Items.Add("GOLFER TOTALS")
    For golfer As Integer = 0 To 3
      lstOutput.Items.Add(golfers(golfer) & ": " & golferTotal(golfer))
    Next
    lstOutput.Items.Add("")
    'Display average per round
    lstOutput.Items.Add("ROUND AVERAGE")
    For round As Integer = 0 To 3
      lstOutput.Items.Add(round + 1 & ": " &
                          FormatNumber(roundTotal(round) / 4))
    Next
  End Sub
```



```
GOLFER TOTALS
Eun Hee Ji: 284
Candie Kung: 285
In-Kyung Kim: 286
Cristie Kerr: 286

ROUND AVERAGE
1: 70.75
2: 72.75
3: 70.50
4: 71.25
```

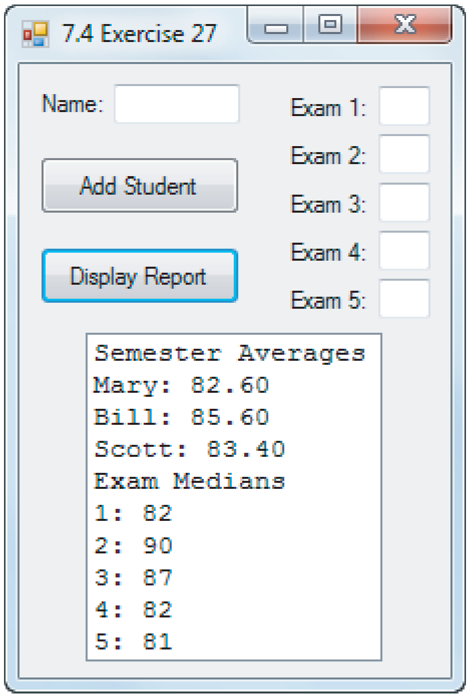


University: Stanford

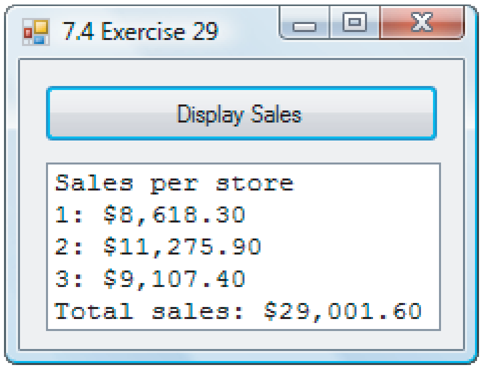
Education Engineering Law

25.
```
  Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
    Dim ranking(2, 4) As String
    Dim disciplines(2) As String
    Dim table() As String = IO.File.ReadAllLines("Ranking.txt")
    Dim data() As String
    For field As Integer = 0 To 2
      data = table(field).Split(","c)
      disciplines(field) = data(0)
      For rank As Integer = 0 To 4
        ranking(field, rank) = data(rank + 1)
      Next
    Next
    Dim result As String = ""
    For category As Integer = 0 To 2
      For rank As Integer = 0 To 4
        If txtName.Text.ToUpper = ranking(category, rank).ToUpper Then
          'Append category name to result
          result &= disciplines(category) & "   "
        End If
      Next
    Next
    If result = "" Then
      txtOutput.Text = "None."
    Else
      txtOutput.Text = result
    End If
  End Sub
```

27.
```
  Dim scores(14, 4) As Integer      'Stores students' exam scores
  Dim count As Integer              'Current number of students stored
  Dim names(14) As String           'Stores students' names

  Private Sub btnAdd_Click(...) Handles btnAdd.Click
    If (count = 15) Then
      MessageBox.Show("Fifteen students already stored.", "Warning")
    Else
      count += 1
      names(count - 1) = txtName.Text
      scores(count - 1, 0) = CInt(txtExam1.Text)
      scores(count - 1, 1) = CInt(txtExam2.Text)
      scores(count - 1, 2) = CInt(txtExam3.Text)
      scores(count - 1, 3) = CInt(txtExam4.Text)
      scores(count - 1, 4) = CInt(txtExam5.Text)
      'Reset input
      txtName.Clear()
      txtExam1.Clear()
      txtExam2.Clear()
      txtExam3.Clear()
      txtExam4.Clear()
      txtExam5.Clear()
      txtName.Focus()
    End If
  End Sub

  Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
    Dim sum As Double, even As Boolean
    lstOutput.Items.Clear()
    lstOutput.Items.Add("Semester Averages")
    For i As Integer = 0 To count - 1
      sum = 0
      For exam As Integer = 0 To 4
        sum += scores(i, exam)
      Next
      lstOutput.Items.Add(names(i) & ": " & FormatNumber(sum / 5))
    Next
    'Display median on the exams
    lstOutput.Items.Add("Exam Medians")
    even = (Int(count / 2) = count / 2)
    For exam As Integer = 0 To 4
      lstOutput.Items.Add(exam + 1 & ": " &
                       Median(scores, count, exam, even))
    Next
  End Sub
```



```
Semester Averages
Mary: 82.60
Bill: 85.60
Scott: 83.40
Exam Medians
1: 82
2: 90
3: 87
4: 82
5: 81
```



```
Sales per store
1: $8,618.30
2: $11,275.90
3: $9,107.40
Total sales: $29,001.60
```

29.
```
Private Sub btnDisplay_Click(...) Handles btnDisplay.Click
  'Load data into an array, cumulate totals, and display a report
  Dim totalSales As Double
  Dim sales(,) As Integer = {{25, 64, 23, 45, 14},
                             {12, 82, 19, 34, 63},
                             {54, 22, 17, 43, 35}}
  Dim price() As Double = {12, 17.95, 95, 86.5, 78}
  'Cumulate totals
  Dim totals(2) As Double
  For store As Integer = 0 To 2
    For item As Integer = 0 To 4
      totals(store) += sales(store, item) * price(item)
    Next
  Next
  'Display report, storing grand total in totals(0)
  lstOutput.Items.Add("Sales per store")
  For store As Integer = 0 To 2
    lstOutput.Items.Add(store + 1 & ": " & FormatCurrency(totals(store)))
    totalSales += totals(store)
  Next
  lstOutput.Items.Add("Total sales: " & FormatCurrency(totalSales))
End Sub
```