Arrays

CE 311 K - Introduction to Computer Methods

Daene C. McKinney

Introduction

• Arrays
• Two-Dimensional Arrays
• String Arrays & Splitting
Arrays

- A group of related things
- Sequentially indexed data structure (matrix)
- All elements in an array have same data type
- An element of an array is accessed using the array name and an index, e.g., A(i)

Array Definition and Initialization

- An array is defined using a declaration statement.

  Dim arrayName(0 To end) As dataType

  - allocates memory for “end” elements
  - Index of first element is 0
    - arrayName(0) is the first element
    - arrayName(end) is the last element
Example

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    Dim A(0 To 4) As Double
    For i = 0 To 4
        A(i) = 1 / (2 * (i + 1))
        ListBox1.Items.Add("A(\(\text{\(} i = \text{\(}\)\)) = \(\text{\(} \frac{1}{(2(\text{\(} i + 1))}\}\))") & vbCrLf
    Next
End Sub
```

Flow Example - Revisited

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
    Dim A(0 To 100) As Double
    Dim sum As Double = 0
    Dim count As Integer = 0
    Do While st.Peek <> -1
        sum += st.ReadLine
        count += 1
    Loop
    st.Close()
    Dim average As Double = sum / count
    Dim sum1 As Double = 0
    For i = 0 To 99
        sum1 += (Math.Pow((A(i)), 2))
    Next
    Dim std dev As Double = Math.Sqrt((sum1 / count) - average ^ 2)
    ListBox1.Items.Add("Average = " & Math.Round(average, 0))
    ListBox1.Items.Add("Std. Dev. = " & Math.Round(std dev, 0))
End Sub
```
Flow Example – Revised Again

```vbnet
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    Dim flow(0 To 100) As Double
    Dim st As IO.StreamReader = IO.File.OpenText("C:\temp\flow.txt")
    count = 0
    Do While st.Peek <> -1
        flow(count) = st.ReadLine
        count = count + 1
    Loop
    st.Close()!
    average = Compute_Ave(flow, count)
    stdev = Compute_Stdev(flow, count, average)
    ListBox1.Items.Add("Average = " & Math.Round(average, 0))
    ListBox1.Items.Add("Std. Dev. = " & Math.Round(stdev, 0))
End Sub
```

Pass Array to function “average” (see next slide)
Pass Array to function “stdev” (see next slide)

Flow Example - Revised

```vbnet
Function Compute_Ave(ByVal flow() As Double, __
    ByVal count As Integer) As Double
    Dim sum As Long = 0
    For i As Long = 0 To count
        sum = sum + flow(i)
    Next
    Return sum / count
End Function
```

Function to compute average

```vbnet
Function Compute_Stdev(ByVal flow() As Double, __
    ByVal count As Long, __
    ByVal average As Double) As Double
    Dim sum As Long = 0
    For i As Long = 0 To count - 1
        sum = sum + (flow(i) - average)^2
    Next
    Return Math.Sqrt(sum / (count - 1))
End Function
```

Function to compute standard deviation
Multi-Dimensional Arrays

Example – 2D Arrays

- Distances between Texas cities in km

<table>
<thead>
<tr>
<th></th>
<th>Austin</th>
<th>Houston</th>
<th>El Paso</th>
<th>Dallas</th>
</tr>
</thead>
<tbody>
<tr>
<td>Austin</td>
<td>0</td>
<td>237</td>
<td>844</td>
<td>283</td>
</tr>
<tr>
<td>Houston</td>
<td>237</td>
<td>0</td>
<td>1081</td>
<td>362</td>
</tr>
<tr>
<td>El Paso</td>
<td>844</td>
<td>1081</td>
<td>0</td>
<td>916</td>
</tr>
<tr>
<td>Dallas</td>
<td>283</td>
<td>362</td>
<td>916</td>
<td>0</td>
</tr>
</tbody>
</table>
2-D Array to Store Contents of Table

Dim mile(0 To 3, 0 To 3) As Double

<table>
<thead>
<tr>
<th></th>
<th>Austin</th>
<th>Houston</th>
<th>El Paso</th>
<th>Dallas</th>
</tr>
</thead>
<tbody>
<tr>
<td>Austin</td>
<td>0</td>
<td>237</td>
<td>844</td>
<td>283</td>
</tr>
<tr>
<td>Houston</td>
<td>237</td>
<td>0</td>
<td>1081</td>
<td>362</td>
</tr>
<tr>
<td>El Paso</td>
<td>844</td>
<td>1081</td>
<td>0</td>
<td>916</td>
</tr>
<tr>
<td>Dallas</td>
<td>283</td>
<td>362</td>
<td>916</td>
<td>0</td>
</tr>
</tbody>
</table>

Text file containing data

Matrix containing data

Example

Declare array

Load data into array before hitting the button

Use data in array after entering origin and destination and hitting the button
Flow Duration Curve

Pr\{Flow ≤ q\} = \frac{i}{n+1} \quad \text{Exceedence Probability}

\text{Pr}\{Flow > q\} = 1 - \frac{i}{n+1}

<table>
<thead>
<tr>
<th>Unssorted Flows</th>
<th>Sorted Flows</th>
<th>Rank</th>
<th>Pr{Flow &lt; q}</th>
<th>Pr{Flow &gt; q}</th>
</tr>
</thead>
<tbody>
<tr>
<td>Million m3</td>
<td>Million m3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10817</td>
<td>6525</td>
<td>1</td>
<td>0.01</td>
<td>0.99</td>
</tr>
<tr>
<td>11126</td>
<td>7478</td>
<td>2</td>
<td>0.02</td>
<td>0.98</td>
</tr>
<tr>
<td>11503</td>
<td>8014</td>
<td>3</td>
<td>0.03</td>
<td>0.97</td>
</tr>
<tr>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
<td>…</td>
</tr>
<tr>
<td>12615</td>
<td>18754</td>
<td>91</td>
<td>0.98</td>
<td>0.02</td>
</tr>
<tr>
<td>16675</td>
<td>20725</td>
<td>92</td>
<td>0.99</td>
<td>0.01</td>
</tr>
</tbody>
</table>

Flow Duration Curve
String Arrays & Splitting

Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    Dim flow(0 To 100) As Double
    Dim count As Long
    Dim sr As New System.IO.StreamReader(C�数templow_{date}.csv"
    Dim line, field(0 To 1) As String

    count = 0
    Do While sr.Peek <> -1
        line = sr.ReadLine
        field = line.Split(",""
        flow(count) = field(1)
        count = count + 1
    Loop
    sr.Close()

    average = Compute_Ave(flow, count)
    stdev = Compute_StDev(flow, count, average)

    ListBox1.Items.Add("Average = " & Math.Round(average, 0)
    ListBox1.Items.Add("Std. Dev. = " & Math.Round(stdev, 0)
End Sub

Summary

- Arrays
- Two-Dimensional Arrays
- String Arrays & Splitting