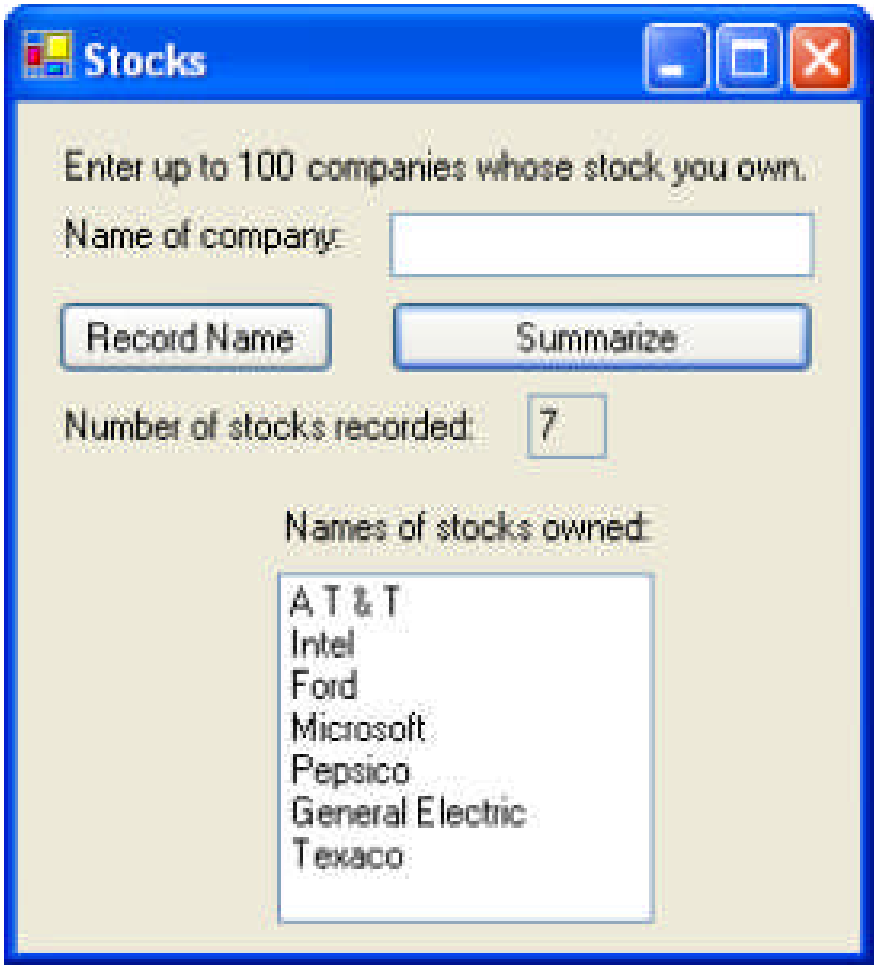
Using Part of an Array

Sometimes we do not know how many elements will be needed in an array.

We can declare a large array, say of 100 elements, and use a counter variable to record the number of elements used.

In this example, the names are an unknown number of companies is placed into an array.

Lab sheet 7.3: Output



In this example, you are required to build a program which stores the company name in a pre-defined array using a **Counter Variable.**

Please refer to lab sheet for detail Instruction.

Lab sheet 7.3: Stocks

Refer to Lab Sheet 7.3 For Detail

In the Program, variable **“Counter”** is a kind of Counter variable which count the usage of the array.

By referencing the counter variable, we can easily know the usage of the array.

Lab sheet 7.3: Code

```
'Demonstrate using part of an array
Dim stock(99) As String
Dim counter As Integer
Private Sub btnRecord_Click(...) Handles btnRecord.Click
  If (counter < 99) Then
    stock(counter) = txtCompany.Text
    counter += 1        'Increment counter by 1
    txtCompany.Clear()
    txtCompany.Focus()
    txtNumber.Text = CStr(counter)
  Else
    MsgBox("No space to record additional companies.", 0, "")
    txtCompany.Clear()
  End If
End Sub
```

Lab sheet 7.3: Code Continued

```
Private Sub btnSummarize_Click(...) _
                    Handles btnSummarize.Click
  'List companies that were recorded
   lstStocks.Items.Clear()
   For i As Integer = 0 To counter - 1
     lstStocks.Items.Add(stock(i))
   Next
End Sub
```

Since the first element of an array is (0), the total number of element of array should be (counter-1)

Ordered Arrays

An array has **ascending order** if [each element] ≤ [next element].

An array has **descending order** if [each element] ≥ [next element].

An array is **ordered** if it has ascending or descending order.

Searching Ordered Arrays

Ordered arrays can be searched more efficiently than unordered arrays. For instance, when searching an array having ascending order, you can terminate the search when you find an element whose value is ≥ the sought-after value.