Applications Programming

Typical Array Operations

Array as a Set of Data

- Position of the data item in the array doesn't represent extra information.
- unsorted array vs sorted array
- Typical operations:
  - process each of the data items (traverse)
  - add a new data item to the array (insert)
  - find a data item (look up)
  - remove a data item from the array (remove)

Look up an element in an unsorted array

```
‘ return the index of the element or -1 if not found
Function LookUp(byRef A() As String, _
                byVal size As Integer, _
                byVal keyword As String) As Integer
    Dim index As Integer
    index = 0
    Do While index < size And A(index) <> keyword
        index = index + 1
    Loop

    If index = size Then
        LookUp = -1
    Else
        LookUp = index
    End If
End Function
```

Look up an element in a sorted array

```
‘ return the index of the element or -1 if not found
' Assuming array A is in ascending order
Function LookUp(byRef A() As String, _
                byVal size As Integer, _
                byVal keyword As String) As Integer
    Dim low As Integer
    Dim high As Integer
    Dim mid As Integer
    low = 0
    high = size - 1
    LookUp = -1
    Do While low <= high
        mid = (low + high) \ 2
        If A(mid) = keyword Then
            LookUp = mid
            low = high + 1 ' End the loop and end the function
        ElseIf A(mid) < keyword Then
            low = mid + 1
        Else ‘ A(mid) > keyword of course
            high = mid -1
        End If
    Loop
End Function
```

Insert an element into an unsorted array

```
‘ check for duplicate first
Function Insert(byRef A() As String, _
                byRef size As Integer, _
                byVal newElem As String) As Boolean
    Dim index As Integer
    index = LookUp(A, size, newElem)
    If index = -1 Then
        A(size) = newElem
        size = size + 1
        Insert = True
    Else
        Insert = False
    End If
End Function
```

Insert an element into a sorted array

```
Function Insert(byRef A() As String, _
                byRef size As Integer, _
                byVal newElem As String) As Boolean
    Dim index As Integer
    index = LookUp(A, size, newElem)
    If index = -1 Then
        index = size - 1
        Do While index >= 0 And A(index) > newElem
            A(index + 1) = A(index)
            index = index - 1
        Loop
        A(index + 1) = newElem
        size = size + 1
        Insert = True
    Else
        Insert = False
    End If
End Function
```

Remove an element from an unsorted array

```
Function Remove(byRef A() As String, _
                byRef size As Integer, _
                byVal Elem As String) As Boolean
    Dim index As Integer
    index = LookUp(A, size, Elem)
    If index = -1 Then
        Remove = False
    Else
        A(index) = A(size - 1)
        size = size - 1
        Remove = True
    End If
End Function
```

Remove an element from a sorted array

```
Function Remove(byRef A() As String, _
                byRef size As Integer, _
                byVal Elem As String) As Boolean
    Dim index As Integer
    index = LookUp(A, size, Elem)
    If index = -1 Then
        Remove = False
    Else
        Do While index < size -1
            A(index) = A(index + 1)
            index = index + 1
        Loop
        size = size - 1
        Remove = True
    End If
End Function
```

Use Row/Column as Array

- Array operations can be extended to rows or columns of data in the worksheet
- start row/column number == index 0
  row/column + 1 == index 1
  ….
  row/column + N == index N