PARTITION OPERATORS IN LINQ

Divide an input sequence into two separate sections without rearranging the elements of the sequence and then returning one of them.

<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
<th>C# Query Expression Syntax</th>
<th>VB Query Expression Syntax</th>
</tr>
</thead>
<tbody>
<tr>
<td>Skip</td>
<td>Skips some specified number of elements within a sequence and returns the remaining ones</td>
<td>Not Applicable</td>
<td>Skip</td>
</tr>
<tr>
<td>SkipWhile</td>
<td>Same as that of Skip with the only exception that number of elements to skip are specified by a Boolean condition</td>
<td>Not Applicable</td>
<td>Skip While</td>
</tr>
<tr>
<td>Take</td>
<td>Take a specified number of elements from a sequence and skip the remaining ones</td>
<td>Not Applicable</td>
<td>Take</td>
</tr>
<tr>
<td>TakeWhile</td>
<td>Same as that of Take except the fact that number of elements to take are specified by a Boolean condition</td>
<td>Not Applicable</td>
<td>Take While</td>
</tr>
</tbody>
</table>

**Example of Skip - Query Expression**

**VB**

```vbnet
Module Module1
    Sub Main()
        Dim words = {"once", "upon", "a", "time", "there", "was", "a", "jungle"}

        Dim query = From word In words
        Skip 4

        Dim sb As New System.Text.StringBuilder()
        For Each str As String In query
            sb.AppendLine(str)
        Next
        Console.WriteLine(sb)
        Console.ReadLine()
    End Sub
End Module
```

When the above code in VB is compiled and executed, it produces the following result:

```
there
was
a
jungle
```

**Example of Skip While - Query Expression**

**VB**

```vbnet
Module Module1
```

Sub Main()
    Dim words = {"once", "upon", "a", "time", "there", "was", "a", "jungle"}

    Dim query = From word In words
        Skip While word.Substring(0, 1) = "t"

    Dim sb As New System.Text.StringBuilder()
        For Each str As String In query
            sb.AppendLine(str)
            Console.WriteLine(str)
        Next
    Console.ReadLine()
End Sub
End Module

When the above code in VB is compiled and executed, it produces the following result:

once
upon
a
was
a
jungle

Example of Take - Query Expression

VB

Module Module1
    Sub Main()
        Dim words = {"once", "upon", "a", "time", "there", "was", "a", "jungle"}

        Dim query = From word In words
            Take 3

        Dim sb As New System.Text.StringBuilder()
            For Each str As String In query
                sb.AppendLine(str)
                Console.WriteLine(str)
            Next
        Console.ReadLine()
    End Sub
End Module

When the above code in VB is compiled and executed, it produces the following result:

once
upon
a

Example of Take While - Query Expression

VB

Module Module1
    Sub Main()
        Dim words = {"once", "upon", "a", "time", "there", "was", "a", "jungle"}

        Dim query = From word In words
            Take While word.Length < 6

        Dim sb As New System.Text.StringBuilder()
            For Each str As String In query
                sb.AppendLine(str)
                Console.WriteLine(str)
            Next
        Console.ReadLine()
    End Sub
End Module
When the above code in VB is compiled and executed, it produces the following result:

```vbnet
Next
Console.ReadLine()
End Sub
End Module
```

Once upon a time there was a