LINQ TIPS
PARTITIONING SETS OF ELEMENTS

numbers *Int[]*	10	6	8	12	3	9	6	18	29	21
Take(4) *IEnumerable<int>*	10	6	8	12						
TakeWhile(i => i <= 13) *IEnumerable<int>*	10	6	8	12	3	9	6			
Skip(7) *IEnumerable<int>*								18	29	21
SkipWhile(i => i>=4) *IEnumerable<int>*					3	9	6	18	29	21
Chunk(4) *IEnumerable<int[]>*	10	6	8	12						
	3	9	6	18						
	29	21								

LINQ TIPS
PARTITIONING SETS OF ELEMENTS

```csharp
// Set of 10 numbers

int[] numbers = { 10, 6, 8, 12, 3, 9, 6, 18, 29, 21 };

// The first 4 numbers

IEnumerable<int> take1 = numbers.Take(4);

// All numbers before the first number greater than 13

IEnumerable<int> take2 = numbers.TakeWhile(i => i <= 13);

// All numbers after the 7th number

IEnumerable<int> skip1 = numbers.Skip(7);

// All numbers after the first number less than 4

IEnumerable<int> skip2 = numbers.SkipWhile(i => i >= 4);

// 4 Number Pack List

IEnumerable<int[]> packs = numbers.Chunk(4);
```