Review

Well done! You've learned a lot of useful information in this lesson:

A list, or `List<T>`, is a generic, sequential data structure. The specific type that it contains is specified upon instantiation.

Lists are effectively unlimited. They "grow" and "shrink" as the number of elements rises and falls.

List values can be accessed by index using square brackets: `[ ]`.

To create an empty list, use a basic constructor. To create a list with values, use *object initialization*.

`Add()` is used to add an element to a list.

`Remove()` is used to remove an element from the list. It returns `true` if it is successful, `false` otherwise.

`Count` is the number of elements in the array.

`Contains()` returns `true` if the argument exists in the list, `false` otherwise.

A sequence within a list is called a *range*. There are specific methods for working with ranges, including `GetRange()`, `AddRange()`, `InsertRange()`, and `RemoveRange()`.

Lists are a type of *generic collection*, which are defined with generic type parameters. The type parameters are specified when instantiating any generic class.

Use lists and dictionaries in your code by including this line at the top of your file:

```
using System.Collections.Generic;
```

If you ever forget the constructors, properties, and methods taught here, you can find them in the [Microsoft documentation for `List<T>`].

Finally, you may be wondering: if lists have so many great features, why did we bother with arrays at all?

The answer is (this an optional part of the lesson): under the hood, lists actually use arrays! When we construct a list, the C# compiler constructs an array and stores the elements there. If the list gets longer than the array's length, the compiler copies the list elements to a new, longer array. To the developer, it just looks like the list is infinitely long.

Thus arrays are faster to use when you have a pre-determined number of elements, and lists are better to use when the number is unknown or you'd like the extra features. If you'd like to dive deeper into this concept, start with the `Capacity` property.

☑Instructions

1.
Add your city to the list, then use a `foreach` loop to print the whole list to the console.