Removing

To remove a specific item from a list we use the `Remove()` method. It expects the specific item as an argument and it returns `true` if it was successfully removed. This code removes `"Delhi"` from the list and returns `true`:

```
List<string> citiesList = new List<string> { "Delhi", "Los Angeles", "Kiev"
};
bool success = citiesList.Remove("Delhi");
// success is true
```

If the specific item does NOT exist in the list, the method call returns `false`.
Since `"Dubai"` isn't in the list, `success` will be `false`:

```
success = citiesList.Remove("Dubai");
// success is false
```

If you remove an element in the middle of the list, all of the elements will be "shifted" down one index. In the first example, the list was originally:

```
[ "Delhi", "Los Angeles", "Kiev" ]
```

After the call to `Remove("Delhi")`, the list becomes

```
[ "Los Angeles", "Kiev" ]
```

☑**Instructions**

1.

The list `marathons` has been initialized for you. Print the second element in the list.

Hint

Lists start with index `0`, so the second item in the list is at index `1`.

2.
Remove the item `143.12` (which is currently the second item in the list) and store the result in a `bool` variable named `removed`.

Hint

Here's an example that removes `"eagle"` from the list `birds`. It also stores the returned value in a `bool` named `success`:

```
bool success = birds.Remove("eagle");
```

3.
Print the second element again and the value of `removed`. The second element should be different and `removed` should be `true`.

Hint

After removing `143.12`, the second element should be `146.73` and `removed` should equal `true`.