

Looping through Lists

Like arrays, we can perform an operation for every element in the list using `for` and `foreach` loops.

With `for` loops, make sure to use `Count` to stay within the bounds of the list.

```
for (int i = 0; i < numbers.Count; i++)
{
    Console.WriteLine(number);
}
```

With a `foreach` loop, the counting is handled for you:

```
foreach (int number in numbers)
{
    Console.WriteLine(number);
}
```

Generally, we prefer `foreach` loops because they require less typing and thus less chance for typos.

If the index is used in the operation — like printing out each index and element together — then we'll use `for` loops.

☒ Instructions

1.

The second `for` loop in the code is used to print out a bib for each runner. Replace it with a `foreach` loop that achieves the same objective.

Hint



Here's an example `foreach` loop that loops through the `birds` list.

```
foreach (string bird in birds)
{
    Console.WriteLine(bird.ToUpper());
}
```

