Goal: Use an array of integers to create a landscape.

The code at the bottom of the page contains two arrays: `heights` stores Int values, and `allCoordinates` stores all coordinates in the puzzle world.

Use the `heights` array to determine how many blocks to stack on each coordinate in `allCoordinates`. To do this, you'll need to access specific `Int` values at each index in `heights`.

Accessing a value at an index

```
var heights = [7,3,2,4]
for i in 1...heights[0]
```

Because the value of `heights` at index `0` is `7`, the `for loop` will run `7` times. Now, what if you want to access a different index for each coordinate? You need to store the index value as a variable and increment it.

Example

```
var index = 0
for coordinate in allCoordinates {
   for i in 1...heights[index] {
      world.place(Block(), at: coordinate)
   }
   index += 1
}
```

Be careful. If the value of `index` is greater than the number of items in the `heights` array, you'll try to access a value that doesn't exist. This will give you an index out of range error. You can prevent this by making sure your `index` value is never greater than `heights.count`, the number of items in your array.

Example

```
if index == heights.count {
   index = 0
}
```

1 Fill in the missing code below to place a stack of blocks of different heights at each coordinate.

2 Notice where the `count` property is used to prevent an array out-of-bounds error.

```
var heights = [7, 8, 3, 6]
let allCoordinates = world.allPossibleCoordinates

var index = 0
for coordinate in allCoordinates {
    if index == heights.count {
        index = 0
    }
    for i in 0...heights[index] {
        // Place a block.
        world.place(Block(), at: coordinate)
    }
    // Increment the index.
    index += 1
}
```