Goal: Remove coordinates from one array while appending to another.

Sometimes you'll want to use an item that you remove from an array. Fortunately, a removed item is stored for a short time, so you can assign it to a variable or append it to another array.

Example

```
var rightColumn = world.column(7)
newArray.append(rightColumn.remove(at: 1))
```

In the code above, the coordinate appended to `newArray` is the one that is also removed from `rightColumn`.

You may have noticed that `rightColumn` is initialized with a method. The `world` instance includes a set of methods that let you quickly create an array containing all coordinates in a column or row.

Calling a method to create an array

```
var row1 = world.row(1)
var column5 = world.column(5)

var topRows = world.coordinates(inRows: [5,6,7])
var allCoords = world.allPossibleCoordinates
```

1. Create one empty array of coordinates, and use a method to create another array of all coordinates in row `2`.
2. Each time the outer loop runs, remove an item from the array and append it to your empty array.
3. Iterate over your empty array, placing an instance of type `Character` at each coordinate.

```
// Create an array of all coordinates in row 2.
var row2 = world.row(2)
// Create an empty array of coordinates.
var charArray: [Coordinate] = []

for i in 1...12 {
    for coordinate in row2 {
        world.place(Block(), at: coordinate)
    }
    // Remove a coordinate and append it to your empty array.
    charArray.append(row2.remove(at: 0))
}
```

```
// Place a character for each coordinate added to your empty array.
for coordinate in charArray {
    world.place(Character(), at: coordinate)
}
```