Goal: Write a function to move forward a certain number of times.

In this puzzle, a new function lets you move across multiple tiles using a single command, reducing the repetition in your code. Using a parameter, you'll specify an input (`distance`) for your function. When you call the function, you'll pass in a value, or argument, for `distance`. For example, in `move(distance: 6)`, `6` is the argument.

The function declaration for `move` is provided below with a `distance` parameter. Use the `distance` value in the function to specify how many times to run `moveForward()`. When you call `move`, pass in the argument for `distance` to run `moveForward()` that number of times.

1. Fill in the function definition, using the `distance` parameter in a loop that calls `moveForward()` a given number of times.
2. If you use a `for` loop, make `distance` the number of times the loop runs. Example: `for i in 1 ... distance {`
3. Solve the puzzle using the `move` function.

```
let expert = Expert()

func move(distance: Int) {
    for i in 1...distance {
        expert.moveForward()
    }
}
```

```
expert.move(distance: 6)
expert.turnRight()
expert.move(distance: 2)
expert.turnRight()
expert.move(distance: 5)
expert.turnLeft()
expert.move(distance: 5)
expert.turnLeft()
expert.turnLockUp()
expert.turnLeft()
expert.move(distance: 3)
expert.turnRight()
```

```
expert.move(distance: 3)
expert.turnRight()
expert.move(distance: 4)
expert.collectGem()
```