Goal: Use a function, a loop, and a condition to collect gems or activate switches.

In this puzzle, every other forward movement might lead to a gem, a switch, or nothing at all. When you run the puzzle, the wireframes show the locations where items might appear. To solve the puzzle, you could write lots of `if` statements, but there's a better way.

Start by breaking the puzzle into its basic patterns. There are three major paths, each with two possible gem or switch locations.

1. Using an `if` statement, define the `collectOrToggle()` function to check the contents of a tile.
2. Below your function definition, call `collectOrToggle()` and other commands to solve the puzzle.

```
func collectOrToggle() {
    if isOnClosedSwitch {
        toggleSwitch()
    }
    else if isOnGem {
        collectGem()
    }
}
```

```
for i in 1 ... 1 {
    moveForward()
    moveForward()
    collectOrToggle()
    moveForward()
    moveForward()
    collectOrToggle()
    turnLeft()
    moveForward()
    moveForward()
    turnLeft()
    moveForward()
    moveForward()
```

```
    collectOrToggle()
    moveForward()
    moveForward()
    collectOrToggle()
    turnRight()
    moveForward()
    turnRight()
    moveForward()
    moveForward()
    collectOrToggle()
    moveForward()
    moveForward()
    collectOrToggle()

}
```