Goal: Use the NOT operator to adjust your character's path when there's no gem on a tile.

The logical NOT operator (`!`) changes the value of a Boolean to its opposite—it *inverts* the value.

For example, if the condition `isBlocked` is *false*, then `!isBlocked` is *true*.

Run the puzzle a few times to see what changes. Notice that there are always four gems, but one is at the bottom of a stairway. When there's no gem on the top platform, the stairs extend from that tile.

Use `!` to determine when your character is NOT on a gem. If not on a gem, turn and collect the gem at the end of the stairs. Otherwise, collect the gem.

```
    // Movement Function with for loop
func program() {
    func turnAround() {
        turnLeft()
        turnLeft()
    }
    func stairs() {
        moveForward()
        moveForward()
    }
for i in 1 ... 4 {
    moveForward()
    if isOnGem {
        collectGem()
    }
    else if !isOnGem {
        turnLeft()
        stairs()
        collectGem()
        turnAround()
        stairs()
        turnLeft()
    }
    else if isBlocked {
        turnLeft()
    }
}
}
// Solve Challenge
program()
```