**1.**

**a.**

public boolean simulate() {

int currentPos = 0;

for (int i=0; i<maxHops; i++) {

currentPos += this.hopDistance():

if (currentPos < 0) {

return false;

} else if (currentPos > this.goalDistance) {

return true;

}

}

return false;

}

**b.**

public double runSimulations(int num) {

int pass = 0;

for (int i=0; i<num; i++) {

if (simulate()) {

pass++;

}

}

return pass/num;

}

4.

public LightBoard(int numRows, int numCols) {

this.lights = new boolean[numRows][numCols];

for (int row = 0; row < numRows; row ++) {

for (int col=0; col < numCols; col ++) {

int rand = (int) (Math.random() \* 10) + 40;

lights[row][col] = rand;

}

}

}

b.

public boolean evaluateLight(int row, int col) {

if (this.lights[row][col]) {

int onCount = 0;

for (int i=0; i<this.lights.length; i++) {

if (lights[i][col]) {

onCount ++;

}

}

if (onCount % 2 = 0) {

return false;

} else {

int onCount = 0;

for (int i=0; i<this.lights.length; i++) {

if (lights[i][col]) {

onCount ++;

}

}

if (onCount % 3 = 0) {

return true;

} else {

return this.lights[row][col];

}

}