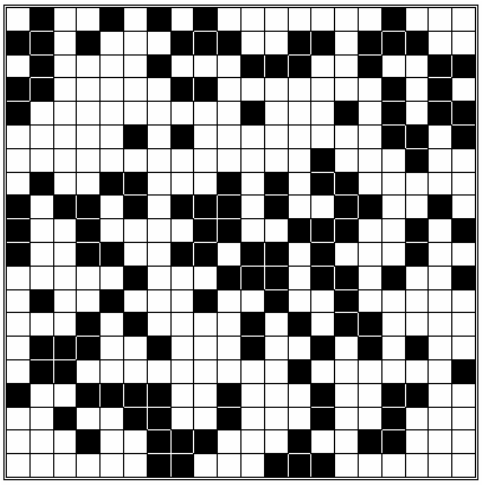
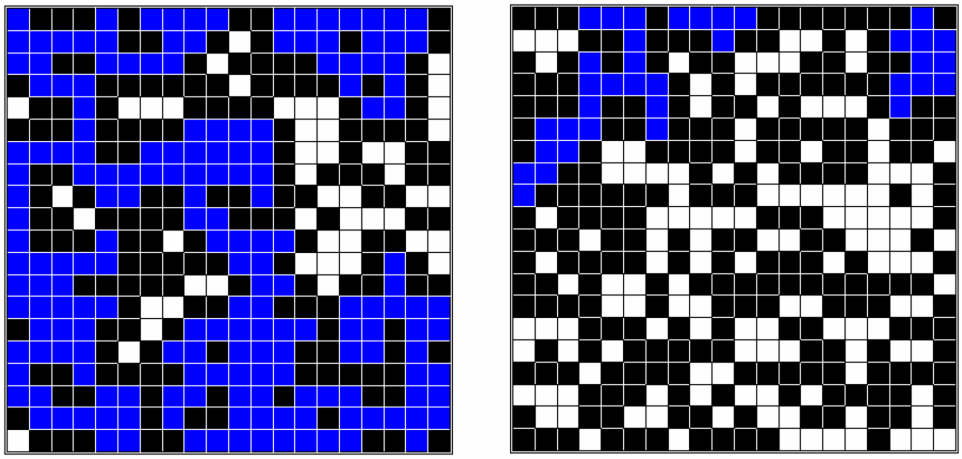
Project 9.3 Percolation

This project assumes that you are familiar with two-dimensional grids from Section 8.2 and have read the part of Section 9.5 on depth-first search.

Suppose we have a grid of squares called *sites*. A site can either be open (white) or blocked (black).



Now imagine that we pour a liquid uniformly over the top of the grid. The liquid will fill the open sites at the top and percolate to connected open sites until the liquid fills all of the open sites that are reachable from an open site at the top. We say that the grid *percolates* if at least one open site in the bottom row is full at the end. For example, the grid below on the left percolates, while the grid on the right does not.



This system can be used to model a variety of naturally occurring phenomena. Most obviously, it can model a porous rock being saturated with water. Similarly, it can model an oil (or natural gas) field; in this case, the top of the grid represents the oil underground and percolation represents the oil reaching the surface. Percolation

systems can also be used to model the flow of current through a network of transistors, whether a material conducts electricity, the spread of disease or forest fires, and even evolution.

We can represent how "porous" a grid is by a *vacancy probability*, the probability that any particular site is open. For a variety of applications, scientists are interested in knowing the probability that a grid with a particular vacancy probability will percolate. In other words, we would like to know the percolation probability for any vacancy probability p. Despite decades of research, there is no known mathematical solution to this problem. Therefore, we will use a Monte Carlo simulation to estimate it.

Recall from Chapter 5 that a Monte Carlo simulation flips coins (metaphorically speaking) at each step in a computation. For example, to estimate the distance traveled by a random walk, we performed many random walks and took the average final distance from the origin. In this problem, for any given vacancy probability p, we will create many random grids and then test whether they percolate. By computing the number that do percolate divided by the total number of trials, we will estimate the percolation probability for vacancy probability p.

Part 1: Does it percolate?

To decide whether a grid percolates, we can use a depth-first search. Recall from Section 9.5 that the depth-first search algorithm completely explores a space by first exploring as far away as possible from the source. Then it *backtracks* to follow paths that branch off. To decide whether a given grid percolates, we must do a depth-first search from each of the open sites in the top row. Once this is done, we simply look at whether any site in the bottom row has been visited. If so, the system percolates. Write a function

```
percolates(grid, draw)
```

that decides whether a given grid percolates. The second parameter is a Boolean that indicates whether the grid should also be drawn using turtle graphics. There is a skeleton program on the book website in which the drawing code has already been written. Notice that some of the functions include a Boolean parameter that indicates whether the percolation should be visualized.

Part 2: Find the percolation probability

For any particular vacancy probability p, we can design a Monte Carlo simulation to estimate the percolation probability:

1. Create a random grid in which, with vacancy probability p, any particular site is open.
2. Test to see if this grid percolates.
3. Repeat steps 1 and 2 a large number of times (say, 10,000), keeping track of the number of grids that percolate.

4. Divide the number that percolate by the total number of trials. This is your estimated percolation probability.

Implement this algorithm with a function

```
percMonteCarlo(rows, columns, p, trials)
```

Part 3: When is a grid likely to percolate?

For what vacancy probability does the percolation probability reach at least 1/2? In other words, what must the vacancy probability be for a system to be more likely than not to percolate?

To answer this question, also write a function

```
percPlot(minP, maxP, stepP, trials)
```

that plots vacancy probability on the x axis and percolation probability on the y axis for vacancy probabilities `minP`, `minP + stepP`, ..., `maxP`. Each percolation probability should be derived from a Monte Carlo simulation with the given number of trials.

You should discover a *phase transition*: if the vacancy probability is less than a particular *threshold* value, the system almost certainly does not percolate; but if the vacancy probability is greater than this threshold value, the system almost certainly *does* percolate. What is this threshold value?