Assignment 4

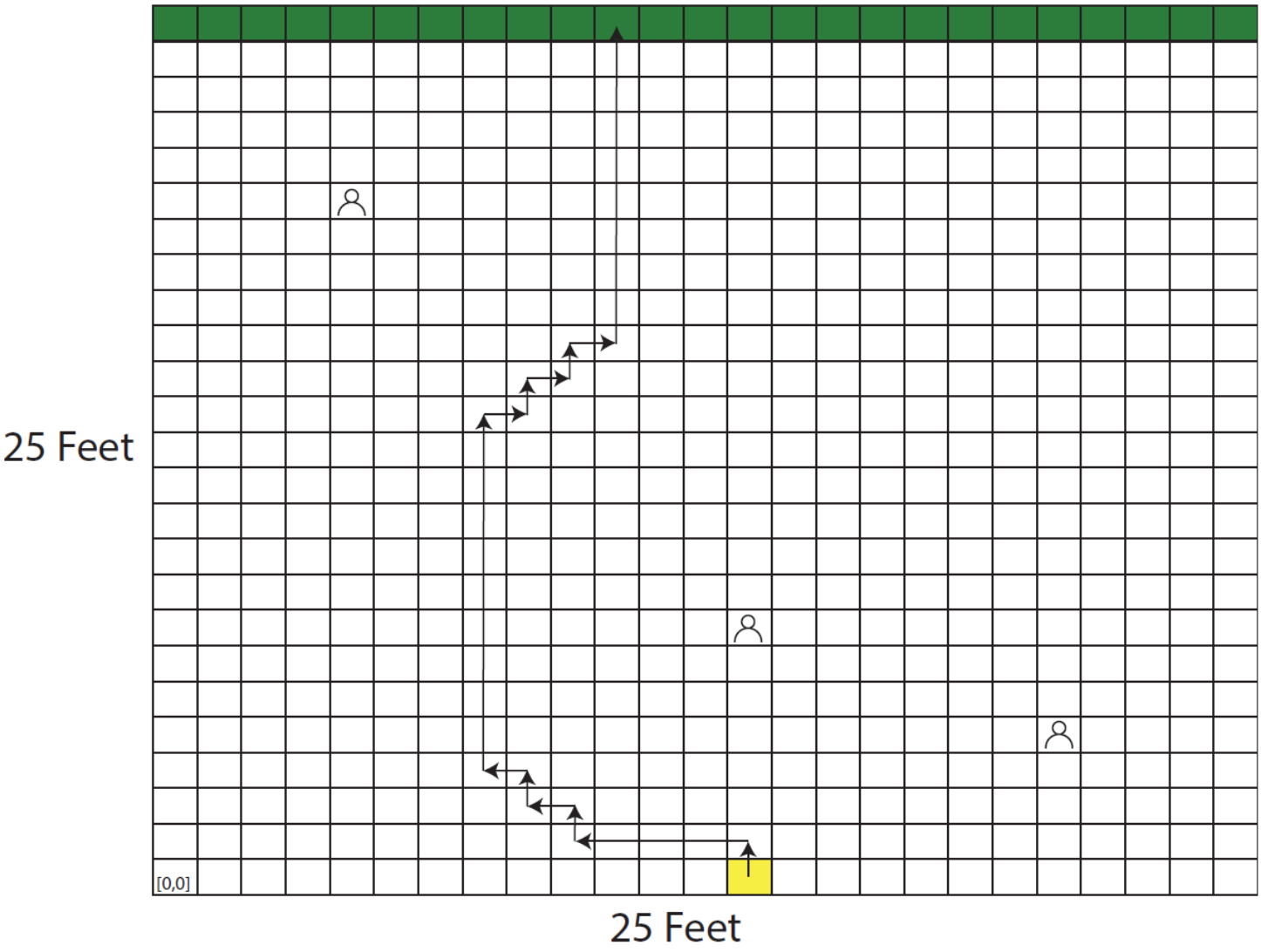
Microproject

Write a Prolog rule which determines if a coordinate is within a certain specified distance of another coordinate on a grid.

Main Project: Social Distancing Simulator

You're trying really hard to adhere to social distancing rules, but sometimes your visits to the local park aren't so easy. You're trying to get from a point on the south side of the park to a nice patch of grass on the north side. There are several people around and sometimes they aren't trying as hard as you to do the right thing. But you've had a great idea! You launch a drone and create a map of the current location of all of the people in the park. Now you just have to write a little program to find a path such that you stay 6 feet away from every other park-goer!

In order to simplify the problem somewhat you divide the part of the park you need to traverse into a 25×25 grid, where each square is 1 foot by 1 foot. You only worry about moving north, east, or west (no looping back to the south). You decide to measure the 6-foot distance from the center of each square using the Pythagorean theorem. Therefore, [0,0] and [0,6] are 6 feet apart, while [0,0] and [4,3] are only 5 feet separated.



Above is a sample configuration of the park. It is possible different positions on the south side of the park might be specified, and the other park-goers might be in different places. The path shown is simply the first path my solution produced – yours may do something different!

In Prolog you will create a program which automatically finds and outputs a path through the park, given a starting position on the south edge of the park, the size of the park, the goal Y coordinate, and the positions of any other people in the park.

```
% The user-facing version of solve takes 5 arguments:
% 1) the starting position on the south edge of the park.
% 2) the ending y-coordinate (on the north edge of the park, typically).
% 3) The size of the grid.
% 4) The locations of the other park-goers.
% 5) The resulting path.
% You will need to have a version of solve with more than 5 arguments as well - this is just the
% Coordinates are represented as [X, Y] lists.
solve([13,0],24,[25,25],[[20,4],[13,7],[4,19]],P).
P = [[13, 0], [13, 1], [12, 1], [11, 1], [10, 1], [9, 1], [9, 2], [8, 2], [8, 3], [7, 3], [7, 4]
```

Constraints + Tips

- **Pressing ';' after an answer results in the "next" possible answer, achieved through backtracking. The program should be able to list all possible answers in this way.**
- Program should work for any legal locations of persons.
- **No use of assert or retract.**
- You may wish to try this on a smaller grid with no persons to start, just to ensure you can find paths. Then slowly scale up.

Development Environment

SWI Prolog (https://www.swi-prolog.org/Download.html)
