The city of Petaluma has asked you to design this year’s Halloween corn maze!!! Using the provided sheets of graph paper, create 2 21 x 31 mazes (Note that a square containing an X represents a wall; a blank square is a path.)

1. Create a maze using the depth-first search method:

This algorithm is a randomized version of the depth-first search algorithm. This approach is one of the simplest ways to generate a maze using a computer. Consider the space for a maze being a large grid of cells (like a large chess board), each cell starting with four walls. Starting from a random cell, the computer then selects a random neighboring cell that has not yet been visited. The computer removes the ‘wall’ between the two cells and adds the new cell to a stack (this is analogous to drawing the line on the floor). The computer continues this process, with a cell that has no unvisited neighbors being considered a dead-end. When at a dead-end it backtracks through the path until it reaches a cell with an unvisited neighbor, continuing the path generation by visiting this new, unvisited cell (creating a new junction). This process continues until every cell has been visited, causing the computer to backtrack all the way back to the beginning cell. This approach guarantees that the maze space is completely visited.

2. Create a maze using the recursive division method:

Mazes can be created with recursive division, an algorithm which works as follows: Begin with the maze's space with no walls. Call this a chamber. Divide the chamber with a randomly positioned wall (or multiple walls) where each wall contains a randomly positioned passage opening within it. Then recursively repeat the process on the subchambers until all chambers are of minimum size. This method results in mazes with long straight walls crossing their space, making it easier to see which areas to avoid. For example, in a rectangular maze, build at random points two walls that are perpendicular to each other. These two walls divide the large chamber into four smaller chambers separated by four walls. Choose three of the four walls at random, and open a one cell-wide hole at a random point in each of the three. Continue in this manner recursively, until every chamber has a width of one cell in either of the two directions.