Goal:	Create a virtual 1-D "board game."
Requirements:	Create a game that has a starting point, an ending point, at least 100 spaces, at least 2 players that move randomly along the board, and at least 25% of the spaces must have a special rule.
Inputs:	• Number of players
Outputs:	• Important turn-by-turn game information
Optional:	• Allow the players to make choices each turn • Make the board cyclical (simple) • Add fancier rules • Create a graphical interface

Background:

  Famous board games such as Candy Land, Snakes and Ladders (Chutes and Ladders is the Milton Bradley version), and Monopoly are made up of spaces that are effectively in a straight line (1D), although Monopoly is a "circle" of sorts. These board games are effectively 1D arrays of "spaces," and each space may or may not contain additional information. In all three games your movement is determined by a random roll of one or more dice, and sometimes the information on a space requires you to move to a different space (and in the case of Monopoly, may trigger a whole set of additional actions).

  The board game in this assignment will most resemble a digital version of Snake and Ladders, an ancient Indian game designed to teach morals (bad behaviors were written on "snake" spaces that caused you to lose progress, while good actions were written on "ladder" spaces that made you jump ahead; the random nature of movement implied that we are not always in control of our situation). This game featured a number of spaces, with players starting on space "1" and winning when they reach the final space.

  This assignment will have you create a Snakes and Ladders-style game using the Python language. Your game must have a board of 100 spaces, at least two players that start on space 1, and the winner is the player that reaches space 100 first. Movement will be through simulated die rolls, and if a player lands on a space with a special rule (a "snake" or a "ladder") they must be transported according to that rule. At least 25% of the spaces on your board should have special rules (morals or other text are optional).

  Each turn, your game must state which player is active, what number they rolled, what the space they land on contains (a rule, or nothing), and finally what space they end their turn on. A player who reaches 100 wins the game, at which point it may end.

  Note: You must decide if players must get to space 100 exactly, or if it is ok to overshoot it and still win!
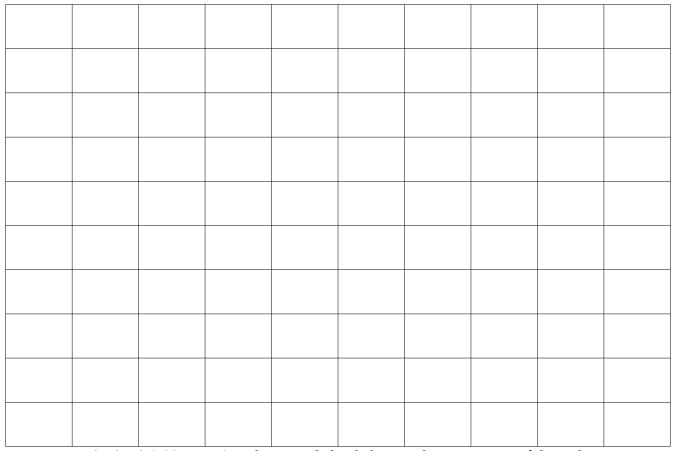
<u>Hints and tips:</u>

In python, the simplest way to make an array of things is to use a **list**. True arrays will be found in the **numpy** (numerical python) module.

If the only special rules on the board are "move forward x spaces" or "move backwards x spaces," you can get away with a list or array that is entirely made of numbers, and the number in the array tells you how many spaces to move (forward if positive, backward if negative). Your program will still need to explain what is happening to the players.

Another option is to fill a list with **functions** ("defs" in python), and when a space is moved to, it calls the function at that place in the list. This technique will give you a lot of power over what can happen on each space.

<u>A note on gameplay:</u> The game created here is ultimately a simple game of chance, with no user control. Most games include an element of choice, where the players can make decisions that will effect the outcome of the game. Sometimes these decisions involve an element of chance, and other times the outcome is always exactly known ("perfect information"), as in chess.

There are multiple ways that you can (optionally) add an element of chance to your game. One way is to allow players to make choices on some spaces. Another way would be to give players a choice every turn, such as how many dice to roll, or the type of dice rolled, or to give them a limited number of "cards" that they can play to alter the outcome of a turn. Your imagination and time are the only limits.

A 10x10 (100 spaces) grid is provided to help you plan your game, if desired.