1.a

Write the recurrence for $V_{\text{minimax}}(s, d)$ in math as a *piecewise function*. You should express your answer in terms of the following functions:

- IsEnd(s), which tells you if s is an end state.
- Utility(s), the utility of a state s.
- Eval(s), an evaluation function for the state s.
- Player(s), which returns the player whose turn it is in state s.
- Actions(s), which returns the possible actions that can be taken from state s.
- Succ(s, a), which returns the successor state resulting from taking an action a at a certain state s.

$$V_{minimax}(s,d) = \begin{cases} Utility(s), IsEnd(s) \rightarrow (Base\ Case: Final\ State) \\ Eval(s), d = 0 \rightarrow (Base\ Case: Terminal\ Depth) \\ \max\limits_{a\ \epsilon\ Actions(s)} \underset{minimax}{\mathrm{V}} (Succ(s,a), d-1) \rightarrow\ Player(s)\ =\ PacMan\ (PacMan\ Move\ -\ Maximize) \\ \max\limits_{a\ \epsilon\ Actions(s)} \underset{minimax}{\mathrm{V}} (Succ(s,a), d) \rightarrow Player(s)\ is\ a\ Ghost\ (Ghost\ Movement\ -\ Minimize) \end{cases}$$

3.a

(a) **[5 points (Written)]** Random ghosts are of course not optimal minimax agents, so modeling them with minimax search is not optimal. Instead, write down the recurrence for $V_{\text{exptmax}}(s, d)$, which is the maximum expected utility against ghosts that each follow the random policy, which chooses a legal move uniformly at random. Your recurrence should resemble that of problem 1a, which means that you should write it in terms of the same functions that were specified in problem 1a.

$$V_{exptmax}(s,d) = \begin{cases} Utility(s), IsEnd(s) \\ Eval(s), d = 0 \\ \max\limits_{a \,\epsilon\, Actions(s)} \underset{minimax}{\mathrm{V}}(Succ(s, \mathrm{a}), \mathrm{d}) \\ \dfrac{\sum_{\mathrm{a'} \epsilon \text{ Actions(s)}} \mathrm{P(a'|s)} \underset{minimax}{\mathrm{V}}(Succ(s, \mathrm{a'}), \mathrm{d\text{-}1})}{|\mathrm{Actions(s)}|} \end{cases}$$

Key points:

So basically, P(a' | s) is the chance that Spirit chooses action a' when in state s. I assume that ghosts act randomly, so that the chance here is evenly distributed across all legal actions. That is to say that each movement has an equal amount, which is 1 divided by the number of actions (|Action(s)) |

Now, V_minimax(Succ(s, a'), d-1) is a mouthful, but it just gets the value Pac-Man after the action is completed. But here's the thing: We're also thinking about what the shadows are going to do to disrupt the Pac-Man cause. We call this the minimum value because Pac-Man tries to increase his score while the shadows try to decrease it. We seek this for a certain depth, d-1.

So since we can capture all possible Pac-Man savings actions, we essentially check what is the smallest average value we can expect. I like to say, “Okay, if Pac-Man tries this move, then that move, and so on, what's the average score we're looking for when considering the ghosts' tricks? This is what we count.