Question 1 - Reflex Agent

1. What is the difference in the definition of a heuristic value for a game state, and for a state in A* search? What properties make a heuristic in either situation 'good'? (3 points total)

An A* search heuristic attempts to estimate value in the context of a path search. (i.e. remaining distance to a goal). For a game state, we are trying to identify the intrinsic value of the state. A good heuristic takes advantage of the available domain-specific information to make the best possible estimate of value of a given state (game state) or for shortest possible path to a destination (A*). Game heuristics just need to give the best possible estimate of value, while A* heuristics also have to obey restrictions like admissibility (and ideally monotonicity).

Question 2 - Minimax

1. Note that Pacman will have suicidal tendencies when playing in situations where death is imminent. Why do you think this is the case? Briefly explain in one or two sentences. (2 points total)

Pacman loses points while moving without eating, it is trying to end the game before losing more points for no benefit.

2. Consider an evaluation function that returns the square of a state's true minimax value. Consider three cases:
   (i) In planning a move (or strategy) for a certain state S, the agent does not hit any terminal state.
   (ii) In planning a move (or strategy) for a certain state S, the agent only hits a terminal state.
   (iii) In planning a move (or strategy) for a certain state S, the agent sometimes hits the terminal state.

Answer the following questions:
   (a) Will the search result in the same strategy in cases (i) and (ii)? [Same/Not Same] (1 point)
   (b) Will the search result in the same strategy in cases (i) and (iii)? [Same/Not Same] (1 point)
   (c) Will the search result in the same strategy in cases (ii) and (iii)? [Same/Not Same] (1 point)

Assume that the minimax values are all positive. Then if the evaluation function returns the square of a state's true minimax value, the relative ordering of the nodes will be the same since this is a monotonic (but non-linear) operator. This is okay, because in minimax (unlike expectimax) we only care about preserving this order.

(a) Same. The nodes will have the same ordering of values resulting in the same strategy.
(b) Not Same. Some terminal states may be hit by (iii), violating monotonicity (e.g. a node with evaluation value 4 (i.e. true value of 2) vs. a terminal node with true value 3)
(c) Not Same. Some terminal states (but not all) may be hit, violating monotonicity. (e.g. a node with evaluation value 4 (i.e. true value of 2) vs. a terminal node with true value 3)

Question 3 - Alpha-Beta Pruning

1. You should notice a speed-up compared to your MinimaxAgent. Consider a game tree constructed for our Pacman game, where b is the branching factor and where depth is greater than d. Say a minimax agent (without alpha-beta pruning) has time to explore all game states up to and including those at level d. At level d, this agent will return estimated minimax values from the evaluation function.
   (a) In the best case scenario, to what depth would alpha-beta be able to search in the same amount of time? (1 point total)

(b) In the worst case scenario, to what depth would alpha-beta be able to search in the same amount of time? How might this compare with the minimax agent without alpha-beta pruning? (2 points total)

(a)
In its best-case Alpha-Beta pruning eliminates branches allowing it to search to depth 2d in the same time. According to the Game Tree Search lecture (Lecture 3 Slide 53), we only need to search $O(b^{d/2})$ nodes, which allows (in theory) searching twice as deep with the same asymptotic time complexity.

(b)
In its worst-case Alpha-Beta pruning does nothing (i.e. no pruning). So, it should behave exactly like regular minimax search with additional overhead for the pruning logic. (this is also consistent with Slide 53). Thus, there may be minor differences in actual run time; however, the asymptotic time complexity should be the same, so it would still be able to search to depth d.

2. True or False: Consider a game tree where the root node is a max agent, and we perform a minimax search to terminals. Applying alpha-beta pruning to the same game tree may alter the minimax value of the root node. (1 point total)

True. Alpha-Beta pruning does not alter the resultant path, but it may alter the resultant value. Let root' be the value after alpha-beta pruning, and root be the value with unadulterated minimax. Then value(root) >= value(root'), but the choice of action will be the same.