**Question 1 Technique & Analysis**

The majority of the code in question 1 is to build an evaluation function for Pacman to pick which legal move it should take. The evaluation function takes into account the following factor: the distance to the closest ghost, the distance to the closest scared ghost, the distance to the closest food, the minimum spanning tree distance to go through all the food, the number of food left and the number of capsule left.

**Question 2 Technique & Analysis**

Question 2 has utilized minimax algorithm, which is to find the best move for a player by assuming the worst move taken by the opponents. Since there might be more than one opponents in the pacman game, the coding in question 2 is slightly different to the standard algorithm provided in class.

It is categorized into 4 functions. The getAction function is to list out all legal action available to pacman at depth 0 and return the one action with the max value by calling the minimax function.

The minimax function is to check whether all the agents in the depth have already made their move. If so, the depth will increment by 1. If the pre-determined depth has been reached or the game has ended, the program will return the evaluation score according to the gameState. It will then determine whether maxValue or minValue should be called for the next agent depending on the nature of this agent.

maxValue function is called and list out all the legal actions by pacman and seek the next value from ghosts by calling minimax function with agentIndex incremented by 1. minValue is in nature the same as maxValue just that it list the actions by ghost and seek the next value from pacman.

Since minimax assumes opponent play an optimal strategy, it will only return an optimal strategy for player if opponent play an optimal strategy yet a suboptimal strategy when opponent do not play an optimal strategy. The time complexity is O(bm) while the space complexity is O(bm).

**Question 3 Technique & Analysis**

Question 3 has utilized minimax and alpha-beta pruning, which is to decrease the number of nodes that are evaluated by the minimax algorithm. The coding in question 3 is based on the minimax algorithm implemented in question 2.

There are 5 functions in the program, 4 of which have already been elaborated in the previous. The additional function bestAction is to carry out the alpha-beta search as stated in the algorithm as well as implement the maxValue node for convenience purpose. Minimax, maxValue and minValue are similar to that in question 2 just that they are now passing two more additional variables, alpha and beta, to record these two variable in order to prune the search tree effectively.

The effectiveness of alpha-beta pruning is highly dependent on the order in which states are examined. If the successors that are likely the best, alpha-beta pruning would be quite effective and it will only need to examine O(bm/2) instead of O(bm). The effective branching factor will be sqrt(b) instead of b.

**Question 4 Technique & Analysis**

Question 4 has utilized expectimax, which is to find the best move for a player by assuming the expected move taken by the opponents by chance. Since there might be more than one opponents in the pacman game, the coding in question 4 is slightly different to the standard algorithm provided in class and based on the coding in question 2.

The function classification by nature is the same as question 2. The expectimax function performs the same service as minimax just that it will call expectValue instead of minValue for opponent agent. expectValue is to return the average value of all legal action available to opponent, which is different to minValue.

expecimax assumes opponent play an random strategy, it will only return an optimal strategy for player if opponent play an random strategy yet a suboptimal strategy when opponent play an optimal strategy.

**Question 5 Technique & Analysis**

Question 5 adopt the same evaluation function as question 1. It also takes into account the following factor: the distance to the closest ghost, the distance to the closest scared ghost, the distance to the closest food, the minimum spanning tree distance to go through all the food, the number of food left and the number of capsule left.

**Question 6 Technique & Analysis**

Question 6 adopt the search function for the pattern that determines the winning state. Since this game has a first mover advantage, the program will seek a list of moves that return a pattern that maintain this first move advantage and pick one of them randomly.