Analysing the Algorithms

1. (1 Mark) How many states did Minimax and Alpha Beta expand for the starting position provided above?

   ANS :
   We found that the starting positions for
   Alpha - Beta expanded — 1414 nodes
   Minimax expanded — 17607 nodes

2. (1.5 Mark) How can player 'O' win the game from a state such as the one shown above?

   ANS :
   If O starts the game as player 2 after the intial player 1.

3. (1.5 Mark) Is it possible for Minimax and Alpha Beta to expand the same number of nodes? Explain why not, or under which conditions this is possible.

   ANS : The output of both methods ought to be the same,
   Alpha-beta, in contrast to minimax, prunes pathways that are certain not to be the best possible states for the current player, that is — max or min.
   This is their fundamental point of distinction. However ,Alpha-beta is a better way to implement minimax.
   This depends on the sequence in which the children are visited. If the children of a node are visited in the worst possible order, pruning might not occur.
   We want to visit the child with the most nodes first to save time by not spending it investigating the other children's worse conditions.
   For min nodes, we want to visit the worst kid first (from our perspective, not the other team's). There are two reliable sources for this information: The static evaluator function can be used to calculate the ranks of the child nodes.
   In earlier searches of the game tree, a large number of game locations were examined using minimaxes (for instance, from previous moves). If available, these data might be used to rank a certain number of nodes.
   When the best child is selected at every opportunity, alpha-beta pruning results in the removal of all the other children from every other level of the tree.
   There is only one youngster under investigation. This results in a significant speedup in the search process, with the tree often able to look twice as deeply as before.