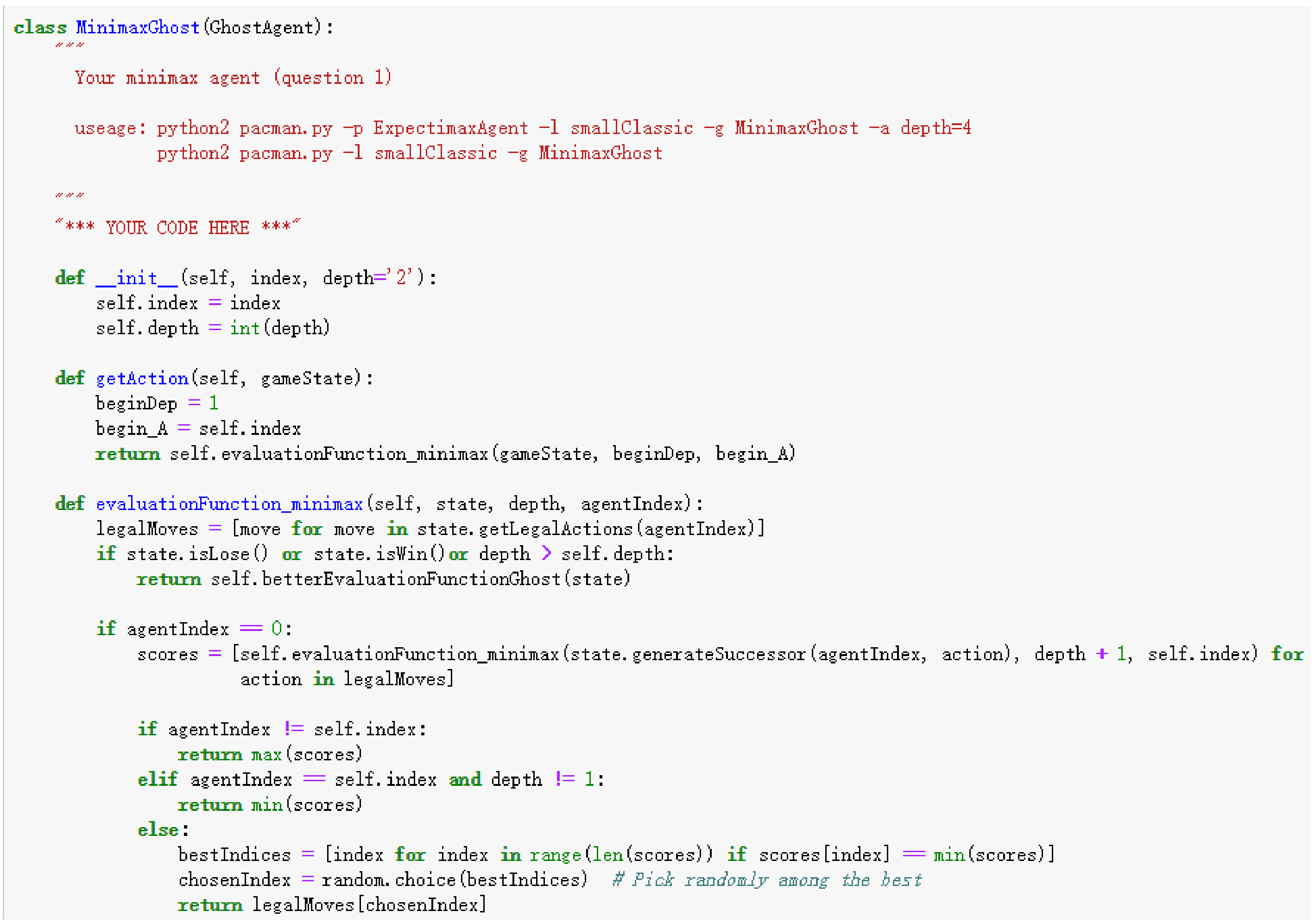
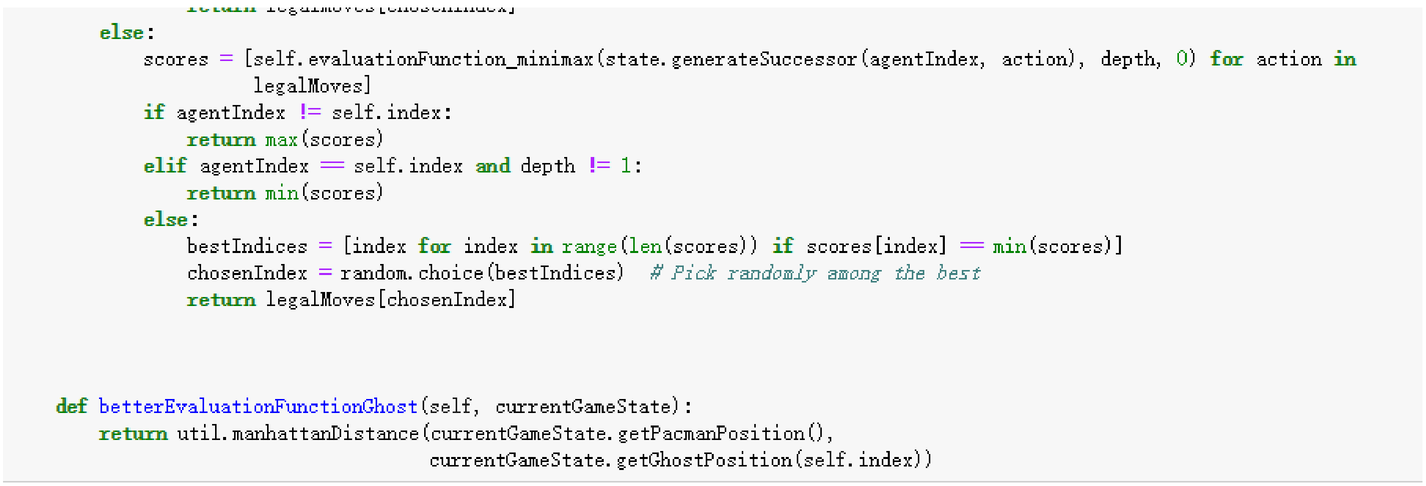
Problem 1



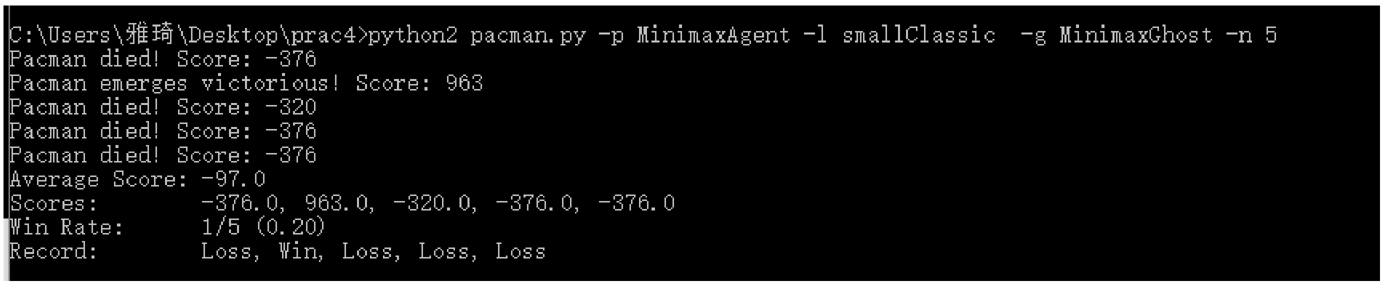
```
class MinimaxGhost(GhostAgent):
    """
      Your minimax agent (question 1)

      useage: python2 pacman.py -p ExpectimaxAgent -l smallClassic -g MinimaxGhost -a depth=4
              python2 pacman.py -l smallClassic -g MinimaxGhost

    """
    "*** YOUR CODE HERE ***"

    def __init__(self, index, depth='2'):
        self.index = index
        self.depth = int(depth)

    def getAction(self, gameState):
        beginDep = 1
        begin_A = self.index
        return self.evaluationFunction_minimax(gameState, beginDep, begin_A)

    def evaluationFunction_minimax(self, state, depth, agentIndex):
        legalMoves = [move for move in state.getLegalActions(agentIndex)]
        if state.isLose() or state.isWin()or depth > self.depth:
            return self.betterEvaluationFunctionGhost(state)

        if agentIndex == 0:
            scores = [self.evaluationFunction_minimax(state.generateSuccessor(agentIndex, action), depth + 1, self.index) for
                      action in legalMoves]

            if agentIndex != self.index:
                return max(scores)
            elif agentIndex == self.index and depth != 1:
                return min(scores)
            else:
                bestIndices = [index for index in range(len(scores)) if scores[index] == min(scores)]
                chosenIndex = random.choice(bestIndices)  # Pick randomly among the best
                return legalMoves[chosenIndex]
```



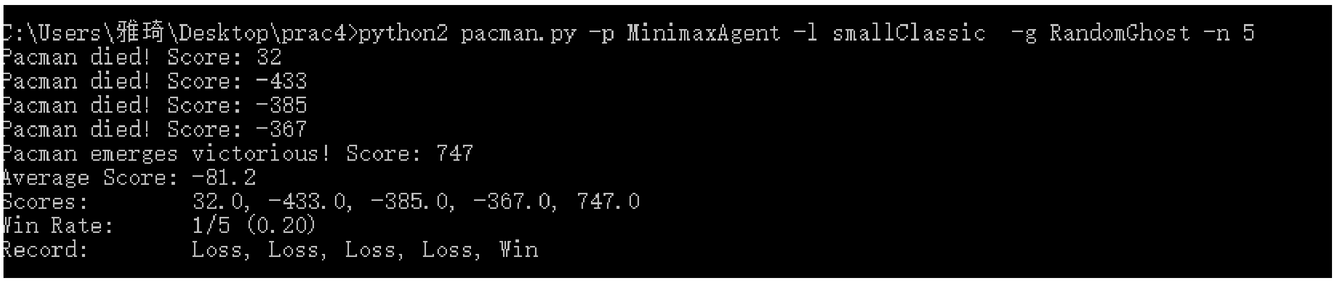
```
        else:
            scores = [self.evaluationFunction_minimax(state.generateSuccessor(agentIndex, action), depth, 0) for action in
                      legalMoves]
            if agentIndex != self.index:
                return max(scores)
            elif agentIndex == self.index and depth != 1:
                return min(scores)
            else:
                bestIndices = [index for index in range(len(scores)) if scores[index] == min(scores)]
                chosenIndex = random.choice(bestIndices)  # Pick randomly among the best
                return legalMoves[chosenIndex]



    def betterEvaluationFunctionGhost(self, currentGameState):
        return util.manhattanDistance(currentGameState.getPacmanPosition(),
                                      currentGameState.getGhostPosition(self.index))
```

Problem 2

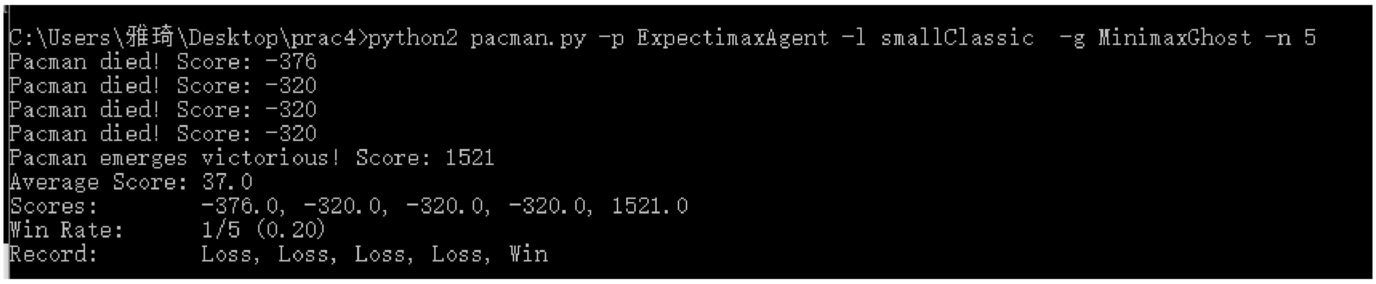
	Adversarial Ghost	Random Ghost
Minimax Pacman	Won 1/5 Avg.Score： -97	Won 1/5 Avg.Score： -81.2
Expectimax Pacman	Won 1/5 Avg.Score： 370	Won 2/5 Avg.Score： 51.2



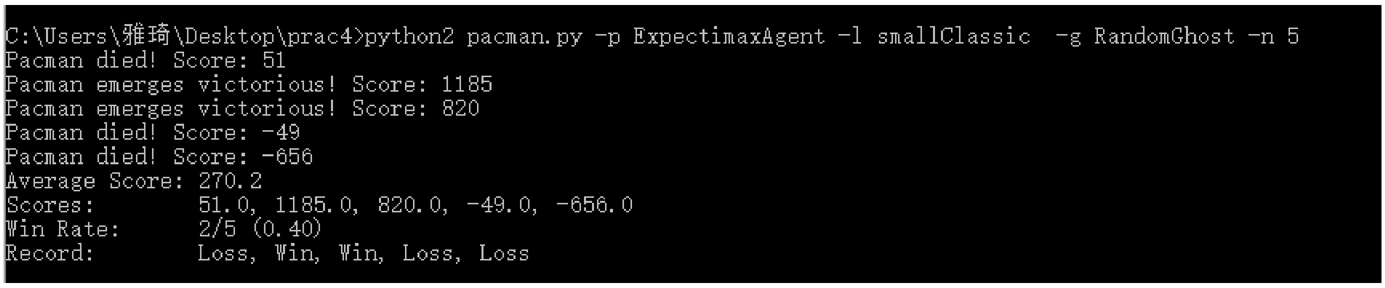
```
C:\Users\雅琦\Desktop\prac4>python2 pacman.py -p MinimaxAgent -l smallClassic  -g MinimaxGhost -n 5
Pacman died! Score: -376
Pacman emerges victorious! Score: 963
Pacman died! Score: -320
Pacman died! Score: -376
Pacman died! Score: -376
Average Score: -97.0
Scores:        -376.0, 963.0, -320.0, -376.0, -376.0
Win Rate:      1/5 (0.20)
Record:        Loss, Win, Loss, Loss, Loss
```



```
:\Users\雅琦\Desktop\prac4>python2 pacman.py -p MinimaxAgent -l smallClassic  -g RandomGhost -n 5
acman died! Score: 32
acman died! Score: -433
acman died! Score: -385
acman died! Score: -367
acman emerges victorious! Score: 747
verage Score: -81.2
cores:        32.0, -433.0, -385.0, -367.0, 747.0
in Rate:      1/5 (0.20)
ecord:        Loss, Loss, Loss, Loss, Win
```



```
C:\Users\雅琦\Desktop\prac4>python2 pacman.py -p ExpectimaxAgent -l smallClassic  -g MinimaxGhost -n 5
Pacman died! Score: -376
Pacman died! Score: -320
Pacman died! Score: -320
Pacman died! Score: -320
Pacman emerges victorious! Score: 1521
Average Score: 37.0
Scores:        -376.0, -320.0, -320.0, -320.0, 1521.0
Win Rate:      1/5 (0.20)
Record:        Loss, Loss, Loss, Loss, Win
```



```
C:\Users\雅琦\Desktop\prac4>python2 pacman.py -p ExpectimaxAgent -l smallClassic  -g RandomGhost -n 5
Pacman died! Score: 51
Pacman emerges victorious! Score: 1185
Pacman emerges victorious! Score: 820
Pacman died! Score: -49
Pacman died! Score: -656
Average Score: 270.2
Scores:        51.0, 1185.0, 820.0, -49.0, -656.0
Win Rate:      2/5 (0.40)
Record:        Loss, Win, Win, Loss, Loss
```

Problem 3

1) random ghost vs. minimax pacman：
   Sometimes the ghosts' moving pattern is like a circle and it's unrelated to the position of pacman because the movement of random ghost is randomly distributed. However, the minimax pacman's movement are in cumulative distribution but sometimes it trapped somewhere and looks silly.
2) random ghost vs. expectimax pacman：
   ghost : uniform distribution .
   expectimax pacman : probability distribution.
3) minimax ghost vs. minimax pacman ：
   inimax ghost : cumulative distribution
   minimax pacman: cumulative distribution.
4) minimax ghost vs. expectimax pacman：
   minimax ghost :cumulative distribution
   the pacman : probability distribution

In case 3) the Pacman agent implementing the correct assumption of the ghosts behavior.

Problem 4

Because the movement of a ghost, will change the gameState and affect evaluation of the minimax tree of another ghost. The ghosts will all chose the movement which can reduce the credits of pacman according to the certain minimax tree.

So, these ghosts will seem as they are cooperating with each other.