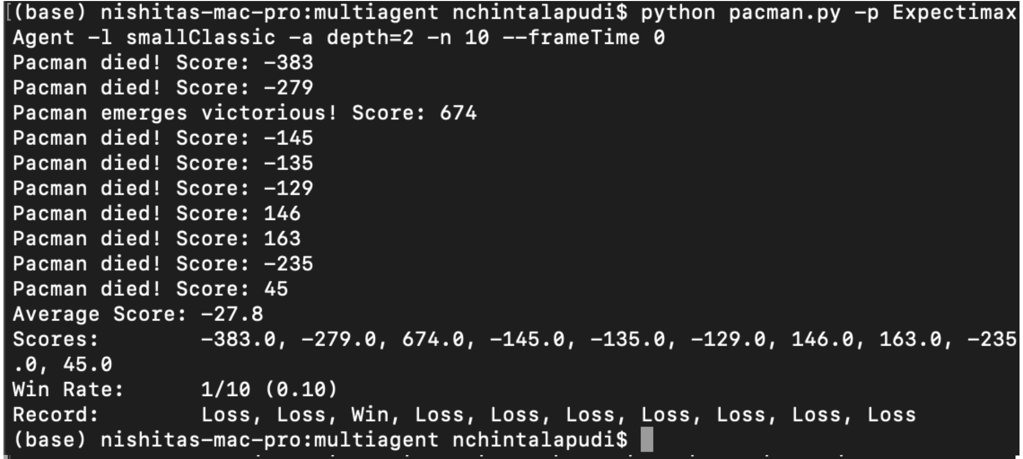
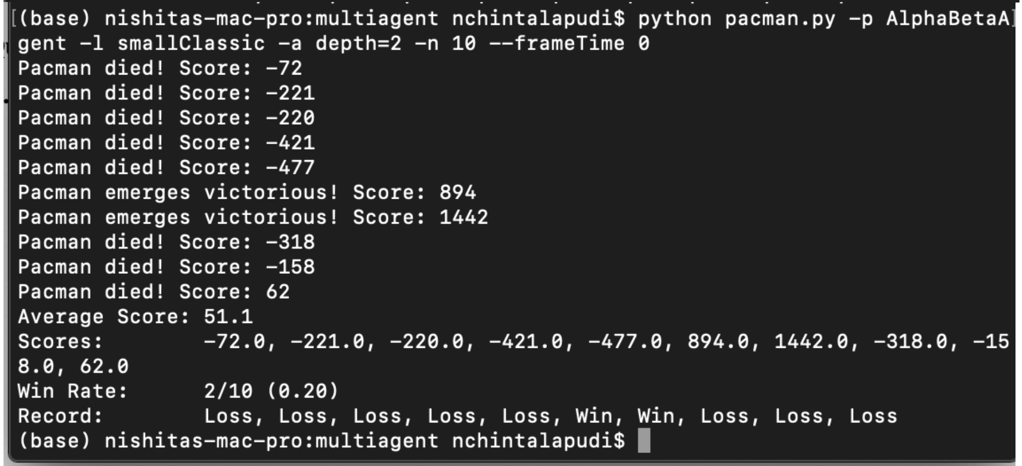
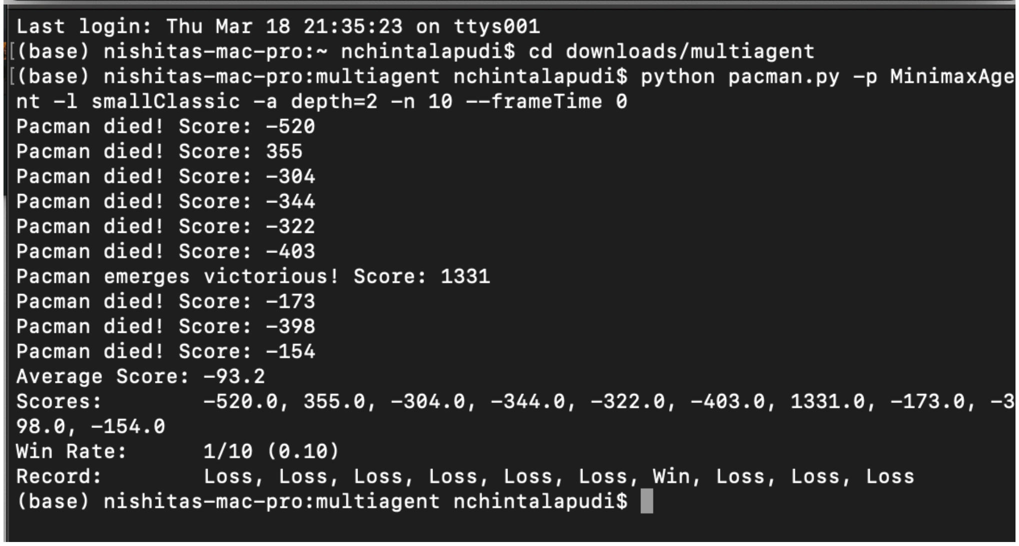
Question 5: report and written questions



```
[(base) nishitas-mac-pro:multiagent nchintalapudi$ python pacman.py -p Expectimax
Agent -l smallClassic -a depth=2 -n 10 --frameTime 0
Pacman died! Score: -383
Pacman died! Score: -279
Pacman emerges victorious! Score: 674
Pacman died! Score: -145
Pacman died! Score: -135
Pacman died! Score: -129
Pacman died! Score: 146
Pacman died! Score: 163
Pacman died! Score: -235
Pacman died! Score: 45
Average Score: -27.8
Scores:        -383.0, -279.0, 674.0, -145.0, -135.0, -129.0, 146.0, 163.0, -235
.0, 45.0
Win Rate:      1/10 (0.10)
Record:        Loss, Loss, Win, Loss, Loss, Loss, Loss, Loss, Loss, Loss
(base) nishitas-mac-pro:multiagent nchintalapudi$ 
```



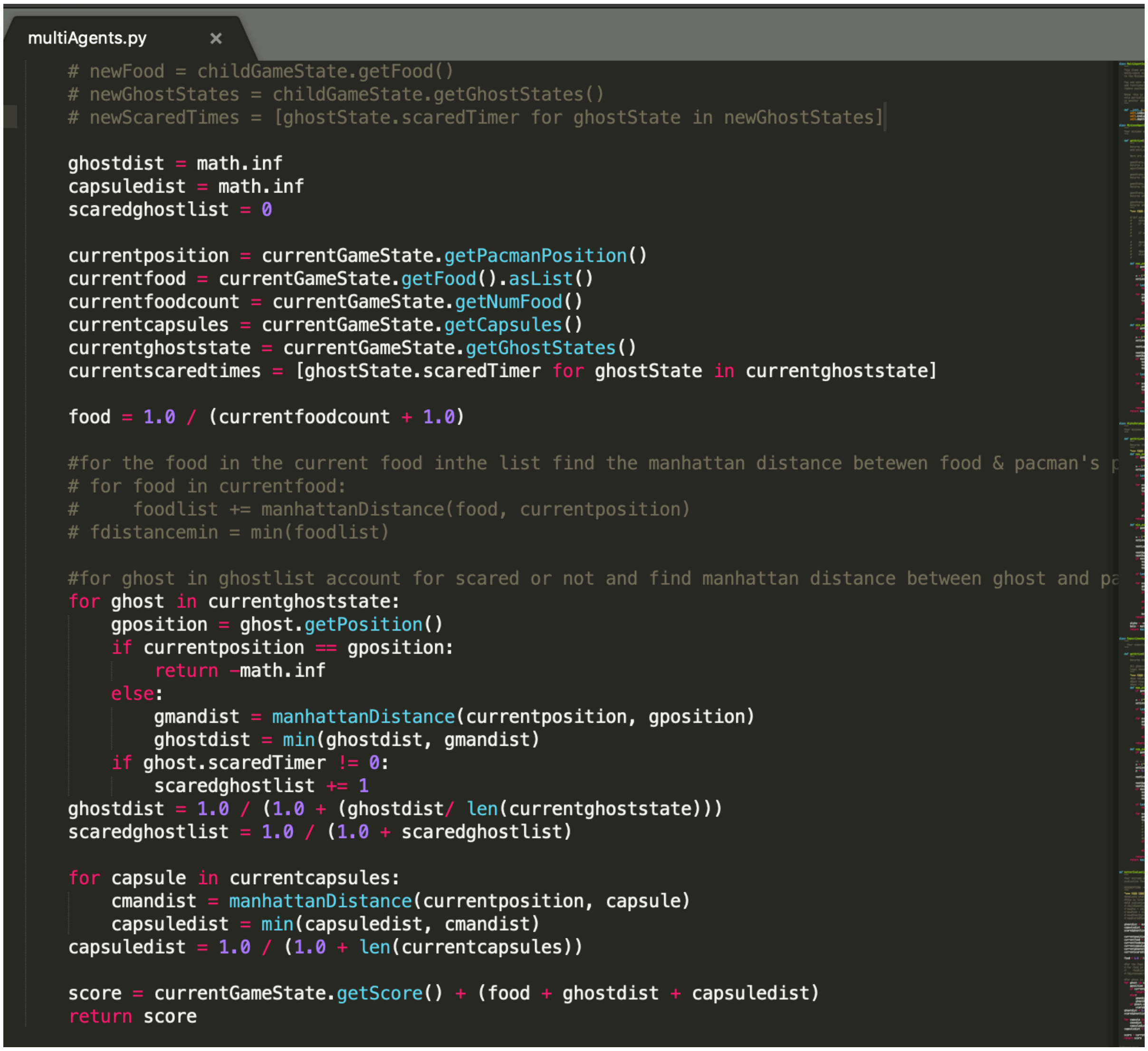
```
[(base) nishitas-mac-pro:multiagent nchintalapudi$ python pacman.py -p AlphaBetaA]
gent -l smallClassic -a depth=2 -n 10 --frameTime 0
Pacman died! Score: -72
Pacman died! Score: -221
Pacman died! Score: -220
Pacman died! Score: -421
Pacman died! Score: -477
Pacman emerges victorious! Score: 894
Pacman emerges victorious! Score: 1442
Pacman died! Score: -318
Pacman died! Score: -158
Pacman died! Score: 62
Average Score: 51.1
Scores:        -72.0, -221.0, -220.0, -421.0, -477.0, 894.0, 1442.0, -318.0, -15
8.0, 62.0
Win Rate:      2/10 (0.20)
Record:        Loss, Loss, Loss, Loss, Loss, Win, Win, Loss, Loss, Loss
(base) nishitas-mac-pro:multiagent nchintalapudi$ 
```



```
Last login: Thu Mar 18 21:35:23 on ttys001
[(base) nishitas-mac-pro:~ nchintalapudi$ cd downloads/multiagent
[(base) nishitas-mac-pro:multiagent nchintalapudi$ python pacman.py -p MinimaxAge
nt -l smallClassic -a depth=2 -n 10 --frameTime 0
Pacman died! Score: -520
Pacman died! Score: 355
Pacman died! Score: -304
Pacman died! Score: -344
Pacman died! Score: -322
Pacman died! Score: -403
Pacman emerges victorious! Score: 1331
Pacman died! Score: -173
Pacman died! Score: -398
Pacman died! Score: -154
Average Score: -93.2
Scores:        -520.0, 355.0, -304.0, -344.0, -322.0, -403.0, 1331.0, -173.0, -3
98.0, -154.0
Win Rate:      1/10 (0.10)
Record:        Loss, Loss, Loss, Loss, Loss, Loss, Win, Loss, Loss, Loss
(base) nishitas-mac-pro:multiagent nchintalapudi$ 
```

1. I do not think they perform well in smallClassic maze. I honestly just think that because it's a smaller maze, it does that thing where it makes the best short-term decision but not really the best decision overall.

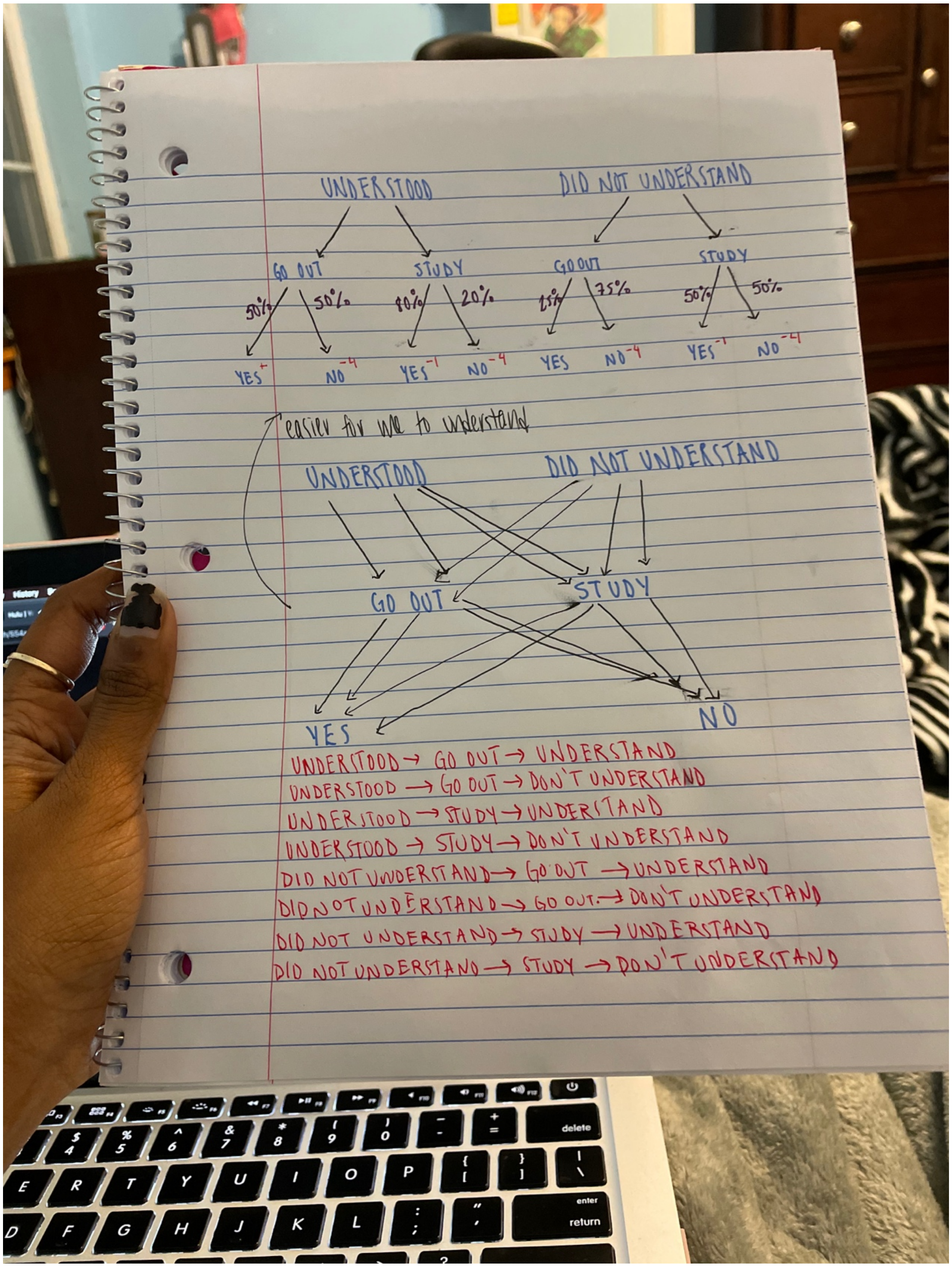
2.



```
# newFood = childGameState.getFood()
# newGhostStates = childGameState.getGhostStates()
# newScaredTimes = [ghostState.scaredTimer for ghostState in newGhostStates]

ghostdist = math.inf
capsuledist = math.inf
scaredghostlist = 0

currentposition = currentGameState.getPacmanPosition()
currentfood = currentGameState.getFood().asList()
currentfoodcount = currentGameState.getNumFood()
currentcapsules = currentGameState.getCapsules()
currentghoststate = currentGameState.getGhostStates()
currentscaredtimes = [ghostState.scaredTimer for ghostState in currentghoststate]

food = 1.0 / (currentfoodcount + 1.0)

#for the food in the current food inthe list find the manhattan distance betewen food & pacman's p
# for food in currentfood:
#     foodlist += manhattanDistance(food, currentposition)
# fdistancemin = min(foodlist)

#for ghost in ghostlist account for scared or not and find manhattan distance between ghost and pa
for ghost in currentghoststate:
    gposition = ghost.getPosition()
    if currentposition == gposition:
        return -math.inf
    else:
        gmandist = manhattanDistance(currentposition, gposition)
        ghostdist = min(ghostdist, gmandist)
    if ghost.scaredTimer != 0:
        scaredghostlist += 1
ghostdist = 1.0 / (1.0 + (ghostdist/ len(currentghoststate)))
scaredghostlist = 1.0 / (1.0 + scaredghostlist)

for capsule in currentcapsules:
    cmandist = manhattanDistance(currentposition, capsule)
    capsuledist = min(capsuledist, cmandist)
capsuledist = 1.0 / (1.0 + len(currentcapsules))

score = currentGameState.getScore() + (food + ghostdist + capsuledist)
return score
```

a. I have three variables defined: one for the ghost the smallest distance away, and same for capsules, and then a list of “scared” ghosts. Then obviously I have a bunch of shortcut variables that I created that are kind of based off of the information from the previous evaluation function.
b. Everything is more or less based off of the capsule’s distance from pacman and the ghost’s distance from pacman. My logic is as follows: it checks through all of the ghosts and if it runs into one, it loses and if not it will set ghostdist to the manhattandistance between ghost and pacman. If the scared timer is not zero, it increments the number of scared ghosts by 1, because that means there is a scared ghost and you should account for it. Pretty much same concept for the capsules. For ghostdist, scaredghostlist, and capsuledist, I do the 1/ (..) thing because the further away the thing is the smaller the reward is supposed to be. Then it increments the current score by adding the other variables accounted for to it, and returns score.

B.



easier for me to understand

UNDERSTOOD → GO OUT → UNDERSTAND
UNDERSTOOD → GO OUT → DON'T UNDERSTAND
UNDERSTOOD → STUDY → UNDERSTAND
UNDERSTOOD → STUDY → DON'T UNDERSTAND
DID NOT UNDERSTAND → GO OUT → UNDERSTAND
DID NOT UNDERSTAND → GO OUT → DON'T UNDERSTAND
DID NOT UNDERSTAND → STUDY → UNDERSTAND
DID NOT UNDERSTAND → STUDY → DON'T UNDERSTAND

i am so tired and mentally ill i cannot do the rest of the math rn i am sorry next time for sure tho