Q1: These agents are called reflex agents because they follow their policy without considering other options or analyzing the environment. The agent does not consider their actions, instead they simply take actions with the better value according to the policy. Value iteration is called offline planning because agent does not do any test runs while creating the policy, unlike qlearning. Values of states are determined by the value of their neighbors instead of test results.

Q2: I lowered the noise to zero, which eventually made falling to chasms impossible. I choose this option because reward on the other side of the bridge was high enough with the current discount rate to be preferred over this side's reward. Because falling into the chasms resulted in a severe penalty, and crossing the bridge increase the risk of falling substantially, I chose to nullify noise to make crossing the bridge more advantageous.

Q3:

```
a) answerDiscount = 0.5 answerNoise = 0 answerLivingReward = -1
```

I assigned the noise zero so going cliffside would have no disadvantage and would therefore be preferable. I assigned discount and living reward high enough for the close exit to be viable over distant exit while going cliffside.

```
b) answerDiscount = 0.5 answerNoise = 0.2 answerLivingReward = -1
```

I assigned noise so going cliffside would be disadvantageous. I assigned discount and living reward high enough for the close exit to be viable over distant exit while going from the long path.

```
c) answerDiscount = 1 answerNoise = 0 answerLivingReward = -1
```

I assigned the noise zero so going cliffside would have no disadvantage and would therefore be preferable. I set discount to 1 so the long-term rewards are as beneficial as short-term, and living reward is set so the longer path is preferable.

```
d) answerDiscount = 1 answerNoise = 0.2 answerLivingReward = -1
```

I assigned noise so going cliffside would be disadvantageous. I set discount to 1 so the long-term rewards are as beneficial as short-term, and living reward is set so the longer path is preferable.

```
e) answerDiscount = 1 answerNoise = 0 answerLivingReward = +100
```

I set the noise to zero so there is no risk of involuntarily going to a terminal state. I set the living reward higher than the rewards of any terminal state, so the agent would never prefer to go to a terminal state.

Q4: Because of the random nature of q learning, states which are more likely to be visited are updated fast while niche conditions are considered less. In our maze this leads to transition values being less than value iteration because the agent is more likely to end in the negative terminal state than the positive one.

Q5: There are too many terminal states in the bridge case which requires more iterations to determine the optimal policy. It is even unlikely for the agent to visit every terminal state in 50 iterations.

Q6: Because each board configuration is considered a separate state and it is impossible for Pacman to explore each of these states and come to general conclusions from his experiences, like discovering that running into ghosts is dangerous for almost all states.

Q7: Pacman can only sense the most imminent dangers, but it is enough in the test maze because there is only one ghost and Pacman cannot be easily stuck between a wall and the ghost. Pacman learns to not get eaten by the ghost, but generally waits for the ghost to come near him to escape.

Pacman does not necessarily try to maximize his score or avoid danger, but with enough training he could perform better.