a.

↑	↑	↑	
↑		↑	→
↑	↑	↑	↑
↑			

Table 1: Optimal policy for VI where $r = 0$,$d = 1$,$p = 1$

10	10	10	
10		10	10
10	10	10	10
10			

Table 2: Final Utilities for VI where $r = 0$,$d = 1$,$p = 1$

→	→	↓	
↓		→	↓
→	→	→	↓
↑			

Table 3: Optimal policy for PI where $r = 0$, $d = 1$, $p = 1$

10	10	10	
10		10	10
10	10	10	10
10			

Table 4: Final Utilities for PI where $r = 0$,$d = 1$,$p = 1$

		↑	↑
↓	↑	↑	↑
→			

Table 5: Optimal policy for Q-learning where $r = 0$, $d = 1$, $p = 1$, $e = 0$, $a = 0.1$, $N = 1000$

	Up	Down	Left	Right
s0	0	0	0	0
s1	0	0	0	0
s2	0	0	0	0
t1	0	0	0	0
s3	0	0	0	0
s4	0.1	0	0	0
s5	0	0	1	0
s6	0	0	0	0
s7	0	0	0	0
s8	0	0	0	0
s9	0	1	0	0
s10	0	0	0	0
t2	0	0	0	0
t3	0	0	0	0
t4	0	0	0	0

Table 6: Resulting Q values for Q-learning where $r = 0$, $d = 1$, $p = 1$, $e = 0$, $a = 0.1$, $N = 1000$

Since there is no cost for moving all the states converged to 10 which is the maximum utility. Optimal policy for all states is to stay in the state because that way the agent can get maximum utility infinitely for value and policy iteration. Resulting Q values and optimal policy is shown above in the related tables. Increasing N does not change anything. Because exploration probability is 0 and agent always takes the best action available at that point; not exploring, just exploiting.

b.

→	→	↓	
↓		→	↓
→	→	→	↓
↑			

Table 7: Optimal policy for VI where $r = -0.01$,$d = 1$,$p = 1$

9.940	9.950	9.960	
9.950		9.970	9.980
9.960	9.970	9.980	9.990
9.950			

Table 8: Final Utilities for VI where $r = -0.01$,$d = 1$,$p = 1$

→	→	↓	
↓		→	↓
→	→	→	↓
↑			

Table 9: Optimal policy for PI where $r = -0.01$, $d = 1$, $p = 1$

0.970	9.950	9.960	
9.950		9.970	9.980
9.960	9.970	9.980	9.990
9.950			

Table 10: Final Utilities for PI where $r = -0.01$,$d = 1$,$p = 1$

		↑	↑
↓	↑	↑	↑
→			

Table 11: Optimal policy for Q-learning where $r = -0.01$, $d = 1$, $p = 1$, $e = 0$, $a = 0.1$, $N = 1000$

	Up	Down	Left	Right
s0	0	0	0	0
s1	0	0	0	0
s2	0	0	0	0
t1	0	0	0	0
s3	0	0	0	0
s4	0.1	-0.001	0	-0.001
s5	0	-0.001	0.99	0
s6	0	-0.001	0	0
s7	0	-0.001	0	0
s8	-0.001	-0.0019	0	0
s9	0	1	0	-0.001
s10	0	0	0	0
t2	0	0	0	0
t3	0	0	0	0
t4	0	0	0	0

Table 12: Resulting Q values for Q-learning where $r = -0.01$, $d = 1$, $p = 1$, $e = 0$, $a = 0.1$, $N = 1000$

So now there is negative reward and it costs the agent -0.01 to move. Since agent wants to maximize utility, will try to avoid going around and try to reach terminal states while

following an optimal policy. Optimal policy at the end of value iteration and policy iteration is the same, the agent is trying to reach the terminal state with a reward of 10, basically all states points to that terminal state. For Q learning, changing N still does not work. Because agent just does the same thing since exploration rate is still 0, finds a way and exploits the best action it learned previously. Since discount factor is 1, the agent does not care about how long it takes to get a reward from a state.

c.

→	→	→	
↓		→	↓
→	↓	→	↓
→			

Table 13: Optimal policy for VI where $r = -0.01$,$d = 0.2$,$p = 1$

−0.004	0.028	0.190	
−0.004		0.068	0.388
0.028	0.190	0.388	1.990
0.190			

Table 14: Final Utilities for VI where $r = -0.01$,$d = 0.2$,$p = 1$

→	→	→	
↓		→	↓
↓	↓	→	↓
→			

Table 15: Optimal policy for PI where $r = -0.01$,$d = 0.2$,$p = 1$

−0.004	0.028	0.190	
−0.004		0.068	0.388
0.028	0.190	0.388	1.990
0.190			

Table 16: Final Utilities for PI where $r = -0.01$,$d = 0.2$,$p = 1$

↑	→	→	
↓		↑	↑
↓	↑	↑	↑
→			

Table 17: Optimal policy for Q-learning where $r = -0.01$,$d = 0.2$,$p = 1$, $e = 0$, $a = 0.1$, $N = 1000$

	Up	Down	Left	Right
s0	-4.21442511e-03	-6.52921560e-03	-5.41365957e-03	-5.97034922e-03
s1	0	0	0	0
s2	0	0	0	0
t1	0	0	0	0
s3	0	0	0	0
s4	1.00000000e-01	-1.00000000e-03	0	-1.00000000e-03
s5	-4.80453164e-03	-1.90000000e-03	1.90000000e-01	-2.64878000e-03
s6	0	-1.00000000e-03	0	-8.59580000e-04
s7	0	-1.00000000e-03	0	0
s8	-1.00000000e-03	-1.90000000e-03	0	0
s9	0	1.00000000e+00	-1.00000000e-03	-1.00000000e-03
s10	0	0	0	0
t2	0	0	0	0
t3	0	0	0	0
t4	0	0	0	0

Table 18: Resulting Q values for Q-learning where $r = -0.01$,$d = 0.2$,$p = 1$, $e = 0$, $a = 0.1$, $N = 1000$

Since discount factor is less than 1, the agent tries to get to the terminal states as fast as possible. Moving costs 0.01 and agent gets 0.2 times of the utility from moving there which is not much considering the terminal states. That's why agent takes the shortest path possible to any terminal states. For Q-learning, agents behavior also changes. The agent wants to get to the closest terminal state as fast as possible so that it can get the reward. This does not produce an optimal policy because of the exploration probability being 0. Agent learns the fastest and best way to maximize utility and applies that policy over and over even though there might be better policy unexplored.

d.
Reward is 5 so that means moving is good for the agent. It will stay in the same state or loop around a few states forever and keep getting 5 reward from doing that infinitely since this is the action that maximizes utility. PI and VI will never converge and the utilities will keep increasing indefinitely.
e.

→	→	←	
↑		↓	↓
→	↑	↑	↓
←			

Table 19: Optimal policy for VI where $r = -0.01$,$d = 1$,$p = 0.5$

9.704	9.724	9.744	
9.684		9.824	9.878
9.690	9.734	9.817	9.926
9.650			

Table 20: Final Utilities for VI where $r = -0.01$,$d = 1$,$p = 0.5$

→	→	↓	
↓		↓	↓
→	→	↑	↓
↑			

Table 21: Optimal policy for PI where $r = -0.01$,$d = 1$,$p = 0.5$

0.780	1.593	2.754	
0.978		5.018	6.907
1.872	3.303	5.359	8.364
1.440			

Table 22: Final Utilities for PI where $r = -0.01$,$d = 1$,$p = 0.5$

		↑	↑
↓	↑	↑	↑
→			

Table 23: Optimal policy for Q-learning where $r = -0.01$,$d = 1$, $p = 0.5$, $e = 0$, $a = 0.1$, $N = 1000$

Since agent goes the way it decides to go only %50 of the time, it takes too much time to converge, at some point the changes become too small for value iteration but the utilities converge close to 10. In policy iteration it takes a lot less iterations to converge but the utilities are not as high as the utilities resulted from value iteration. The resultant policies from the algorithms are different. The optimal policy does not make sense because movement of the agent is stochastic, the utilities depends on probability too much.
f.

→	→	↓	
↓		→	↓
→	→	→	↓
↑			

Table 24: Optimal policy for VI where $r = -0.01$,$d = 0.9$,$p = 0.8$

3.787	4.327	4.941	
3.433		6.211	7.304
3.923	4.635	5.747	8.469
3.192			

Table 25: Final Utilities for VI where $r = -0.01$,$d = 0.9$,$p = 0.8$

→	→	↓	
↓		→	↓
→	→	→	↓
↑			

Table 26: Optimal policy for PI where $r = -0.01$,$d = 0.9$,$p = 0.8$

0.620	2.452	4.370	
1.884		6.055	7.269
3.323	4.539	5.724	8.466
2.563			

Table 27: Final Utilities for PI where $r = -0.01$,$d = 0.9$,$p = 0.8$

i. Due to the transition being stochastic the final utilities are different which leads to different policy. Agent still does not explore more states, just goes with the best policy at that moment.
ii. Changing N does not make a difference in final policy but utilities change.
iii. Changing e to 0.1 didn't change anything in our results but it is supposed to make agent choose a random action over a action that provides best utility for %10 of the time. The result may not changed because the random action may be the best action for that time. iv. Agent does not learn gradually but takes the experience immediately.