CSE 4309 – 001 Machine Learning
Assignment 7

Task – 1

```
value_iteration('environment2.txt', -0.04, 1, 20)
```
utilities:
0.812 0.868 0.918 1.000
0.762 0.000 0.660 -1.000
0.705 0.655 0.611 0.387

policy:
> > > o
^ X ^ o
^ < < <

```
value_iteration('environment2.txt', -0.04, 0.9, 20)
```
utilities:
0.509 0.650 0.795 1.000
0.399 0.000 0.486 -1.000
0.296 0.254 0.345 0.130

policy:
> > > o
^ X ^ o
^ > ^ <

Task – 2

The reward of non-terminal states in chess for reinforcement learning algorithm would be kept something very low, so that the longer sequences are less costly. This will allow our model to go through more states since the less cost would mean that it can find out more sequences in the game, which will result in a more knowledgeable model.

The discount factor will be kept high if we don't want to achieve our results quickly so that the model has enough time to explore other states in chess. However, while playing the game, we would want the model to have a low discount factor so that the agent focus on near term to win the game.

Task – 3

a. R = -0.04 and $\gamma=0.9$

$U(2,2) = 0.8 * (\text{going up}) + 0.1*(\text{left}) + 0.1(\text{right})$

$= 0.8 * (0.9^0 (-0.04) + 0.9^1 (1)) + 0.2*(\text{blocked state})$

$= 0.8 * (-0.04 + 0.9) + 0.2* U(2,2)$

$0.8\, U(2,2) = 0.8 * 0.86$

$U(2,2) = 0.86$

b. U(2,2) = 0.8*(1*r + 0.9) + 0.2*U(2,2)
0.8*U(2,2) = 0.8(r + 0.9)
U(2,2) = r + 0.9
U (2,2) was optimal when it was 0.86 from part a, so take U (2,2) >= 0.86
0.86 >= r + 0.9
- 0.04 >= r
Or r < -0.04