Submission Guidelines
Your submission must be your original work. Do not indulge in any kind of plagiarism or copying. Abide by the honour and integrity code to do your assignment.

You submission must include:

- A legible PDF/Doc document with all your answers to the assignment problems/ questions.
- A folder named as ‘code’ containing the scripts for the assignment along with the other necessary files to run your code.
- A README.txt file explaining how to execute your code.

Naming Convention:
Name the ZIP file submission as follows:
YourName_emailid.zip

This is an evaluation assignment where you need to implement an algorithm from the Reinforcement Learning domain called the value iteration algorithm.

Code: Python

Description: Create a grid environment of 5*5 as shown in the below figure

	1	2	3	4
5	6	7	8	9
10	11	12	13	
14	15	16	17	18
19		20	12	

There are two kinds of grid positions in this grid environment viz. Terminal states and non-terminal states. The terminal states are marked as dark boxes. The non terminal states are S={1,.....,21}. There are four actions in each state that are all possible four directions of movement i.e. North, South, East and West. When an agent takes an action in the grid environment it changes the state of the agent except for those which will take it off the grid.

The value function of a state S can be given as follows

$$V_{\Pi}(s) = \sum_{a} \Pi(a|s) \sum_{s'} p_{ss'} [r + \gamma V_{\Pi}(s')]$$

Where the value of the state is recursively defined using the value of s'

Parameters:

1. Reward in each state is +1 except the terminal state.
2. The probability of all directions (N, S, W, E) = (0.2, 0.3, 0.25, 0.25).

Question 1: Approximate the value function for the given scenario

1. Initialize the value function randomly between (0 to 1) except the terminal state.
2. Approximate the value function at different rates of the discount factor.

Question 2: Approximate the value function

1. Initialize the value function to 0.
2. Approximate the value function at different rates of discount factor.

References:

1. Reinforcement Learning: An Introduction (Sutton and Barto)
2. https://towardsdatascience.com/reinforcement-learning-demystified-solving-mdps-with-dynamic-programming-b52c8093c919