ARTIFICIAL INTELLIGENCE
ASSIGNMENT II
MARKOV DECISION PROCESS

A sample grid world with its properties is given below (input):

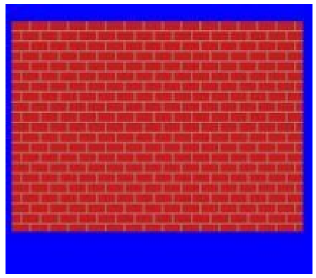
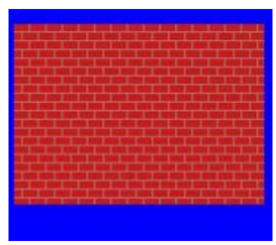
Input:

- First line of the input consists of 2 space separated integers **N M,** which are the dimensions of the grid world.
- **N** lines follow, each having **M** real numbers specifying the rewards for getting into that state.
- Next line consists of 2 space separated integers **E W**, number of end states and number of walls.
- Next **E** lines follow - each having 2 integers - the coordinates of **end states**.
- Next **W** lines follow - each having 2 integers - the coordinates of **walls**.
- Next line has 2 integers specifying the coordinates of the **start state**.
- Next line has 1 integer specifying the **unit step reward**.

Input for the grid world mentioned below is:

```
4 4
0      0    0     X
X/10   0    0     0
0      0    0     0
0      0  -X/5    0
3 2
0 0
0 3
3 2
0 1
2 2
3 0
-X/10
```

The grid world (MDP) to be considered while doing **Part B, C** of the assignment:

+ X/10 (Goal State)			**+X (Goal State)**
			
START		**- X/5 (End State)**	

Grid

- Rows are numbered 0,1,2,3 from top to bottom and columns are numbered 0,1,2,3 from left to right. Eg. Start cell is (3,0)
- The cells (0, 0) and (0, 3) are the positive(green) sinks while cell (3,2) is the negative(red) sink with rewards as mentioned in the figure.
- The blue cells are blocked (assume them as walls).

- The borders of the grid are also walls.

Movement

- Agent can go North, South, East or West.
- Action from a state results in
  - Movement in intended direction with probability 0.8.
  - Movement in directions perpendicular to the intended direction with 0.1 probability each (0.8 + 0.1 + 0.1 = 1). Eg. If action is North, then actual movement will be to North with 0.8 prob, to East with 0.1 prob, & to West with 0.1 prob.
- If an action results in reaching a cell with wall, agent will remain in the same cell.
- No action needs to be performed at terminal states.

Questions

PART A

Write a program (in python) that performs **Value Iteration(VI) Algorithm** on an input grid world (as mentioned above) and outputs the utility map of the states after each iteration of the **VI algorithm** till convergence(along with the utility map after convergence). For convergence/termination, consider at most **1% difference** in each state utility between iterations. (**Assume discount factor = 0.99**)

Output Format:

If the input grid world is of size **n** x **m**, output should consist of **n** lines, each having **m** spaced values. Each value should be rounded off to 3 decimal positions.
Example output format for a grid of size 3 x 2(for each iteration):
5.223 4.667
1.446 3.132
9.636 8.312
//new line after each iteration

PART B

Perform the **VI algorithm** on the below mentioned cases with **X** being your team number.

1. Discount factor:(keep step cost = -X/10; same as above for the below decay rate)
    a. 0.1
    b. 0.99 (same case as the sample input)
2. Step cost: (decay rate = 0.99 is a constant for all the different step costs given below)
    a. X
    b. -X/5
    c. -X/4
    d. -X

(Any parameter not mentioned above, has the same value as given in the sample input)
For each new policy made, carefully analyze it and make a report. (It can contain information on why the policy is in that particular way. How are the parameters influencing the final policy, ...).
Submit the 6 final policies (policies for the above mentioned 6(1.a, 1.b, 2.a, 2.b, 2.c, 2.d) use cases) along with their corresponding observations and conclusions in a pdf.

PART C

Model the above problem(specific MDP) using LP shown below

$$\max(\mathbf{rx}) \mid \mathbf{Ax} = \boldsymbol{\alpha},\ \mathbf{x} \geq 0,$$

1. Model the parameters r, A and α.
2. Use the excel LP solver to compute the x values and the expected utilities for this MDP Please verify that the expected utility obtained is equivalent to the one obtained using the VI algorithm.
Verify that the expected utility obtained is equivalent to the one obtained using the VI algorithm.

Deliverables

You are supposed to submit :

- A working code of Value Iteration (Part A) - **20 marks**
- PDF (Part B) - **12 marks**

- Excel file or .ods (LibreOffice) file, showing the LP solved - **20 marks**

Points to be noted

- You need to load the solver add-in in Excel if not already installed. It comes loaded with LibreOffice packages by default.
- Policies are different from utility maps. Make sure you submit the correct thing for the required questions.
- Any type of plagiarism will result in serious penalties.
- **Submission format: *teamNumber*_Assignment2.zip.** Inside the zip folder, the structure should be:
    - *teamNumber*_Assignment2
        - main.py
        - Report.pdf
        - mdp.ods (or) mdp.xls
- **Submission date: March 11th, 2019, 8 A.M.**