**Programming Assignment on Markov Decision Process (45 points)**

**First Name1: \_\_\_\_\_\_\_\_\_\_\_\_ Last Name1: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**First Name2:\_\_\_\_\_\_\_\_\_\_\_\_ Last Name 2: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

In this problem you will implement the value iteration algorithm for finding the optimal policy for each state of an MDP using Bellman’s equation. Your program should assume as input a file that contains a description of an MDP. Below is a sample input file:

s1 5 (a1 s1 0.509) (a1 s2 0.491) (a2 s1 0.31) (a2 s3 0.69)

s2 10 (a1 s1 0.4) (a1 s2 0.3) (a1 s3 0.3) (a2 s2 0.5) (a2 s3 0.5)

s3 -5 (a1 s1 0.3) (a1 s2 0.3) (a1 s3 0.4) (a2 s1 0.2) (a2 s2 0.8)

Each line in this file stores information for one state in the given MDP. For instance, the first line stores information about state s1: the reward associated with s1 is 5, on action a1 we move to state s2 with probability 0.491 and stay in s1 with probability 0.509, and on action a2 we move to state s3 with probability 0.69 and stay in s1 with probability 0.31. The remaining lines of the file can be interpreted in a similar fashion.

After each of the first 20 iterations of the value iteration algorithm, your program should print the J value and the optimal policy for each state of the given MDP. Hence, the output of your program may look something like:

After iteration 1: (s1 a1 0.225) (s2 a3 5.456) (s3 a3 -2.435)

After iteration 2: (s1 a2 1.224) (s2 a3 4.456) (s3 a1 -1.888)

After iteration 3: (s1 a2 2.345) (s2 a3 4.252) (s3 a1 -1.098)

. . .

The above output says that after the first iteration, s1’s optimal action is a1 and the associated J value is 0.225, s2’s optimal action is a3 and the associated J value is 5.456, and s3’s optimal action is a3 and the associated J value is -2.435.

Your program should allow exactly four arguments to be specified in the invocation of your program: (1) the number of states of the MDP, (2) the number of possible actions, (3) the input file as described above, and (4) the discount factor . No other arguments are allowed.

There should be no graphical user interface (GUI) of any kind. Any program that does not conform to the above specification will receive no credit. It may be helpful to take a look at the sample input and output files attached.

Submit via D2L your source code and a README file that contains instructions for running your program.

**Submission:**

* **Rule1:**
  + If you work with a partner, please name your zipped file as follows:

PA6\_LNAME1\_LNAME2.Zip for folder, i.e., the file name should include both LAST NAMEs.

* + If you work on your own, the format should be

PA6\_LNAME.Zip for folder.

* **Rule2:**
  + **EVERYONE** in the class should submit this Assignment, which should provide all files (like data files etc.. ) that are necessary for the execution of code in the submission folder.