AI-1 - Ali Ul Haq - 500667810

Question 1:

Concisely explain why it is that the TableDrivenAgent is unable to accomplish this task.

Because the TableDrivenAgent is unaware of its whole enviroment, it only percepts the current state and performs actions with the actiontable. If the perception in the actiontable is not available the agent does not know which action it can take. Furthermore it is only limited by the grid size because actions outside that grid size are not taken into account.

Question 2:

Is your ModelBasedAgent guaranteed to eventually clean all rooms in a 3x3 grid environment? If so, show why. If not, show why not. Note: please do not solve the problem by modifying the TableDrivenAgent to choose random actions. This will eventually visit all the rooms, but doesn't really get at the point of the assignment.

Yes it will clean every room for a 3x3. Because when the program is run it will first locate the origin which is alway (0,0) for this grid environment for every points which it is randomly set to. After that it will travel along the x - axis to its maximum grid size. After that it will travel upwards to aquire the GridSize and go to its starting position to start cleaning. After it has reached the starting position the start flag, which has been programmed, will be set to True. In the table below the column represent the x-axis and the rows represent y-axis of the Grid

start = False

Gridpoints			
	0	1	2
0			Start
1			↑
2	Origin →	→	↑

Start = true

Gridpoints			
0	↓	←	Start
1	→	→	↓
2	Origin	←	←

When it has reached its origin again the start flag will be set to False and the process can start over again until the final step has been reached.

Question 3:

Is your ModelBasedAgent guaranteed to be optimal? In other words, does it clean all rooms in the minimum possible number of moves? If so, prove that this is the case. If not, explain what limitations of your agent prevent it from solving the problem optimally. (It's not required that your agent be optimal - just that you indicate whether it is or not.)

No, because my modelbased will follow a spiral pattern downwards from the upmost right point (the point reached when it excecuted the countgrid function). This is a predetermined pattern programmed, it would be ideal if the program would scan the enviroment when it was going to its origin point and remembering the enviroment state. This would allow the program to intelligently clean and thus take a shorter path. The path which my program takes follows a loop and follows that path “stupidly”.

Question 4:

Now let's make the problem more challenging: build a ModelBasedAgent that can solve the NxMVacuumEnvironment for other values of N and M. Notice that the values of N and M are NOT available to the agent; the only way it can figure out how large the grid is, is through exploration. In other words, you cannot modify the NByMVacuumEnvironment; just the agent.

See Class NByMVacuumEnvironment.