# ECEN 689-605/700 Dependable Learning Systems Fall 2018
# Homework 3

Due 1pm U.S. central time, September 28, Friday
Submission must be through ecampus
No late submission accepted!

(20 points) Please implement a linear gradient-descent Q(λ) system with binary features consisting of an agent and an environment. The environment is specified by the model in homework 1, including the state transition probabilities and reward functions. Please note the agent does not know the probabilities and reward functions in advance. When your agent take an action at certain state, you can simulate the environment to obtain the reward and the next state. Please implement in C/C++, java or Python. Finally, please report the Q(s,a) values for all (s,a) pairs, the number of visits for each (s,a) pair for each setting, brief discussion on the results and the source code. The parameters are:

- ε=0.1
- learning rate α=0.2
- discount factor γ=0.5, λ=0.5
- m=3, initialize $\vec{\theta} = (1,1,1)^T$
- Features are $\vec{\phi}_{s_1,a_1} = (0,1,1)^T$, $\vec{\phi}_{s_1,a_2} = (0,0,1)^T$, $\vec{\phi}_{s_2,a_1} = (1,0,1)^T$, $\vec{\phi}_{s_2,a_2} = (1,0,0)^T$, $\vec{\phi}_{s_3,a_1} = (1,1,0)^T$, $\vec{\phi}_{s_3,a_2} = (0,1,0)^T$.
- Each episode starts with $s_0$, and end with $s_H$, which is the terminal state 0 reward, and the subscript here indicates time step. Thus, the agent will take H actions in each episode.
- Please run K episodes. The first episode initializes with $s_1$ defined in homework 1, the 2nd episode starts with $s_2$, … the fourth episode starts with $s_1$, and so on.
- Please generate results for the following two settings
  - H=9, K = 9
  - H=27, K = 27