**Cairo University**

**Faculty of computers and Artificial Intelligence**

**Reinforcement Learning**

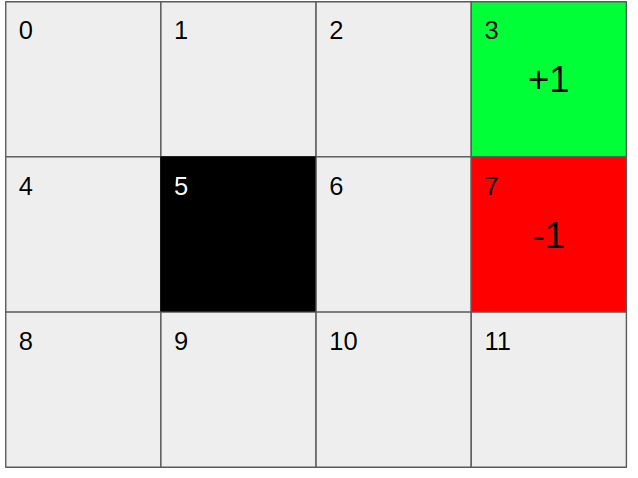
Assignment 0

* **Assignment objective**

The objective of this assignment is to learn:

* + Understanding of a given RL problem parameters.
  + The implementation of the environment.
  + The implementation of Value iteration and q values.
* **Assignment Rules**
  + Due date is 19/3.
  + The assignment will be delivered in groups of 4 - 5 students.
  + Your file name should be 1ID\_2ID\_3ID\_4ID.py
  + Any cheats will got zero.
* **Assignment Description**

Implement value iteration and q-values methods on the Grid World problem.



1. Build the Environment.

\* Number of states is 12 (each grid represent state)

\* Number of actions is 4 (Up, Down, Left, Right)

\* Reward of grid 3 is 1, for grid 7 is -1 and grid 5 is a wall

\* Discount factor (Gamma) = 0.9, noise = 0.2, Num of iterations = 100

1. For every iteration print the Grid Values

Example of the output:

| -0.01 | -0.01 | 0.782 | +1 |

| -0.01 | WALL| -0.01 | -1 |

| -0.01 | -0.01 | -0.01 | -0.01 |

1. After the model converges or go through all iterations extract the policy and print it

Example of the output:

| Right | Right | Right| +1 |

| Up | WALL | UP | -1 |

| Up | Right | UP | Down|

Tip: for building the environment you could use 3\*4 2d array and each state is represented by the index of that array (state 1 => [0][0], state 2 => [0][1] and so on), and for the actions you could write it as Actions = [(1, 0), (0, -1), (-1, 0), (0, 1)] -> Down, Left, Up, Right then the next state when taking action a can be obtained by adding the current state indexes and the action values Ex. We are in state [0][1] and we go down which is (1,0) so the new state will be [0+1][1+0] -> [1][1]. (You can use this implementation or do it your way)

Hint: Any trial will be appreciated so please try.