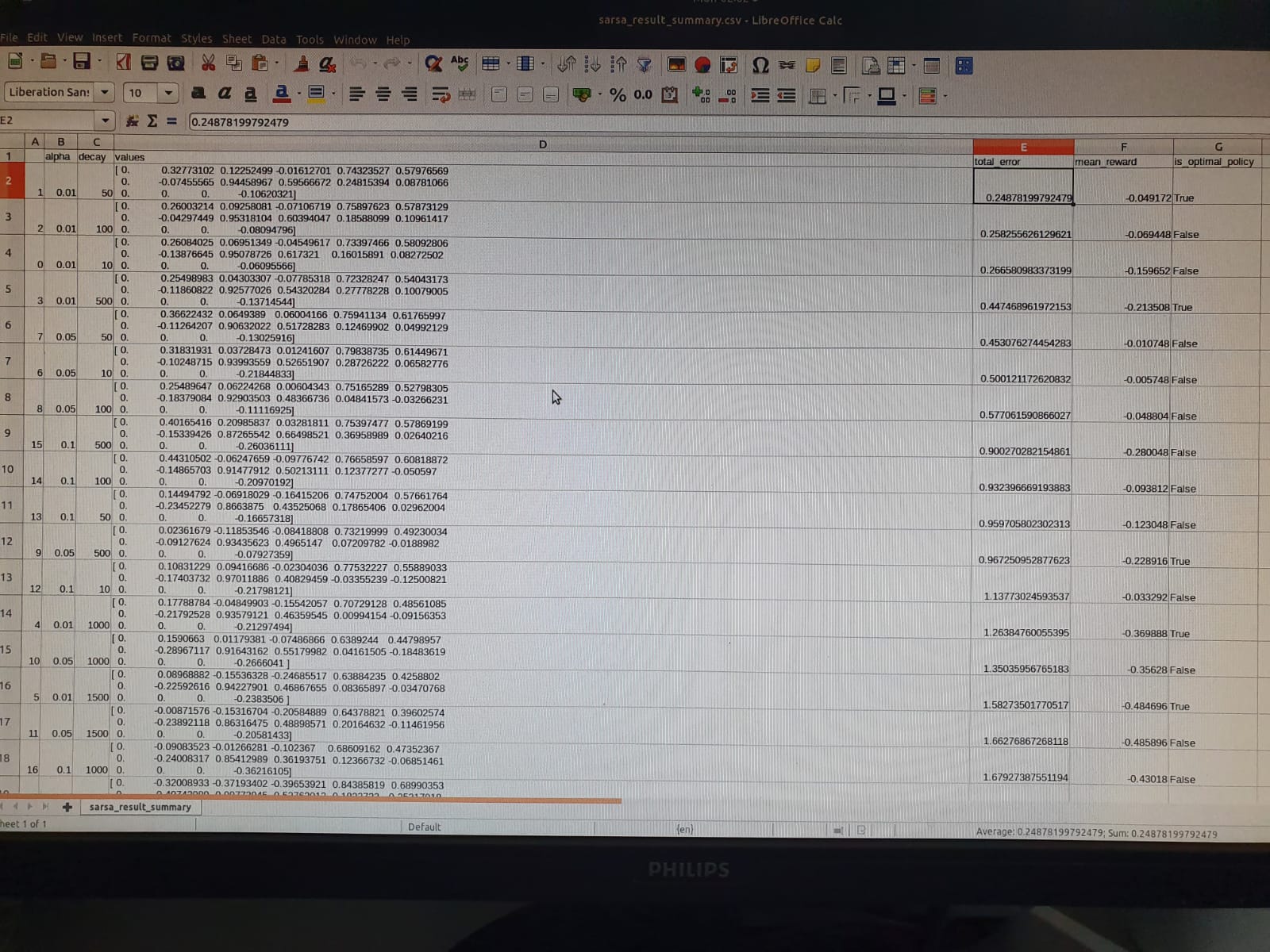
**EXE 2 – sarsa and qlearning**

Question 1 - sarsa:

* The function that implements the algorithms called sarsa and it’s located in world.py
* The optimal hyper parameters for sarsa algorithm are alpha=0.01 and decay=50
* The way I choose the optimal parameters was that I built a function called find\_optimal\_params and for each combination of parameters I run 10K episodes and for each combination I calculated the total deviation from the optimal values in the MDP solution and checked if the policy after 10K was the optimal policy of the MDP. I choose 10k because more than that it takes a lot of time to run all the parameter combinations.

The file with all the combinations and their scores called sarsa\_result\_summary.csv

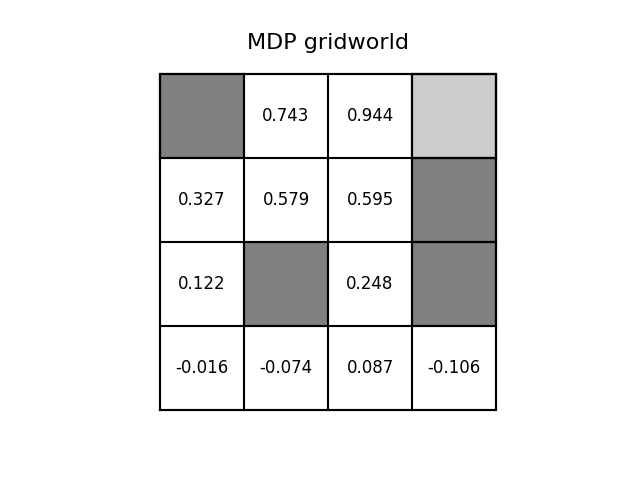
Here are the result of the different combinations after 10K episodes:

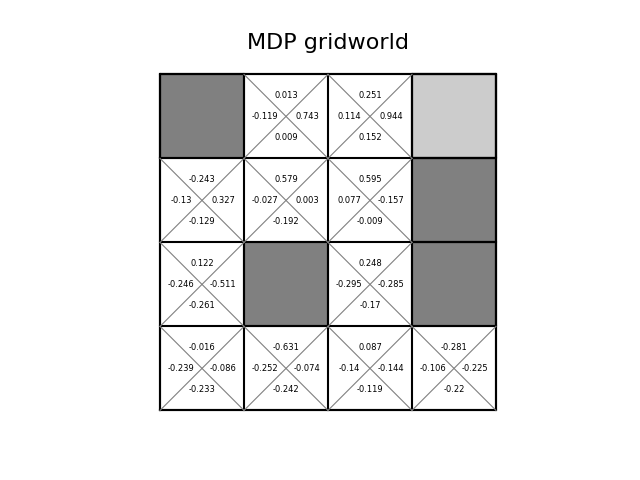


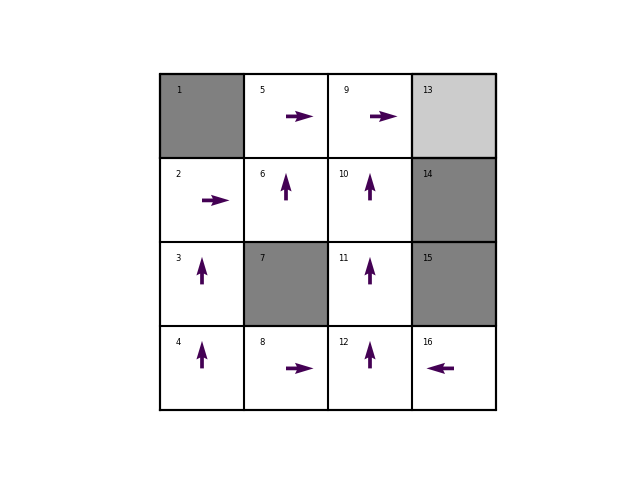
My criteria for best parameters was minimum total error and have same policy as in the optimal MDP solution

Results of the best sarsa combination after 10k episodes:

Values:



Action values

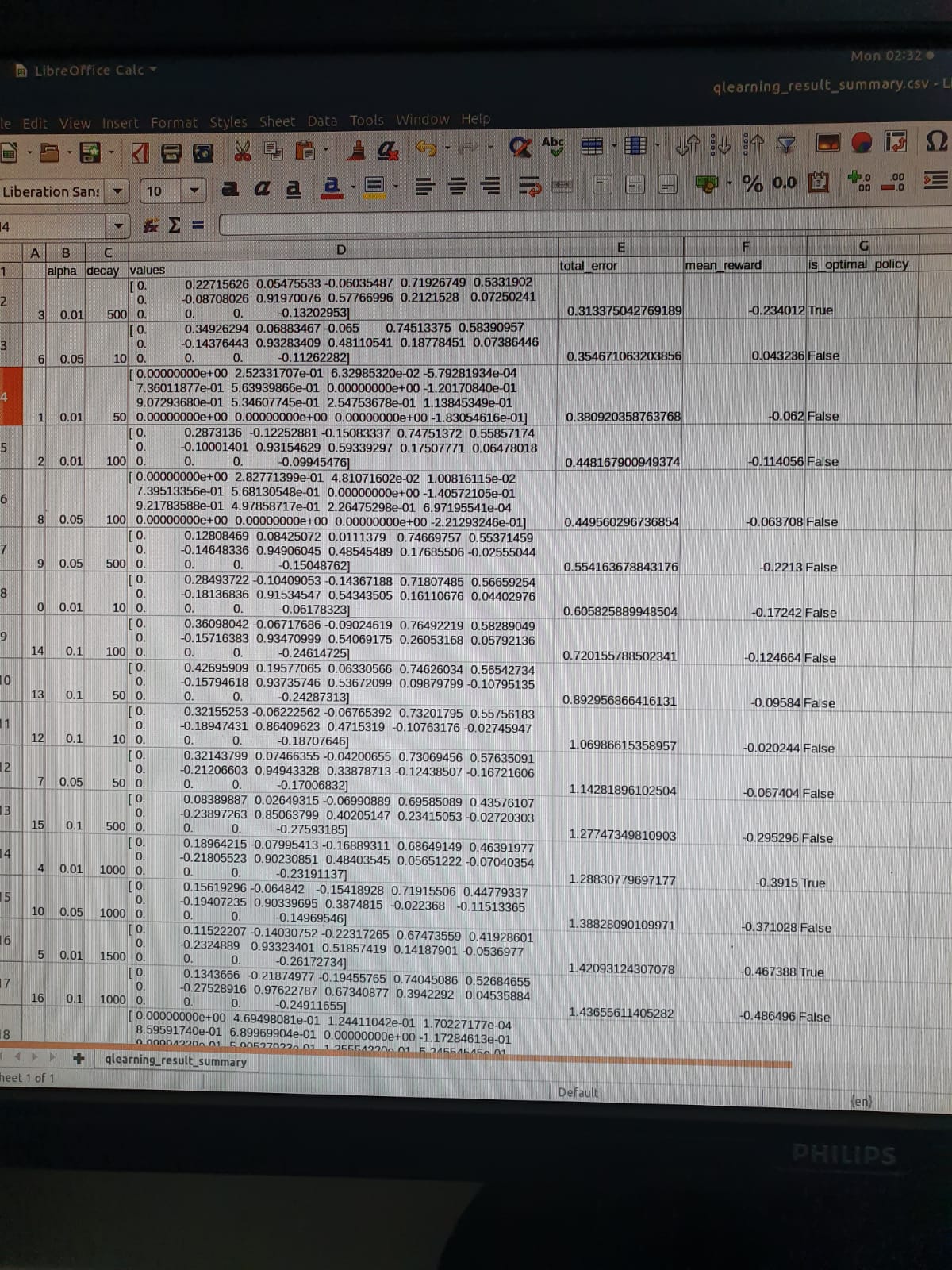
Policy

Question 2 – Qlearning:

* The function that implements the algorithm called Qlearning and it’s located in world.py
* The optimal hyper parameters for Qlearning algorithm are alpha=0.01 and decay=500
* The way I choose the optimal parameters was the same as in sarsa algorithm.

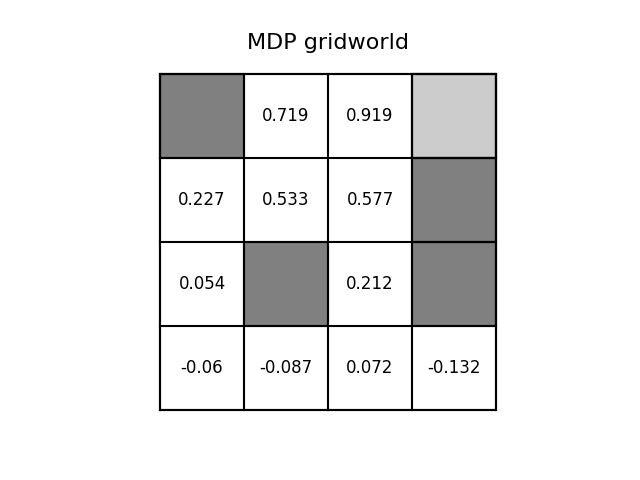
The file with all the combinations and their scores for Qlearning called qlearning\_result\_summary.csv

Here are the result of the different combinations after 10K episodes:

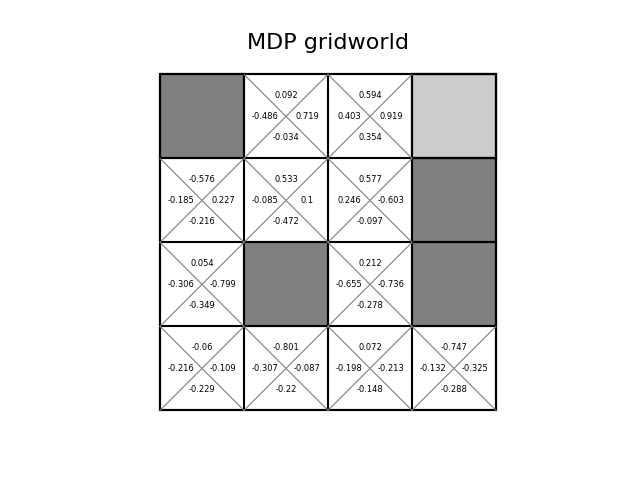


Results of the best Qlearning combination after 10k episodes:

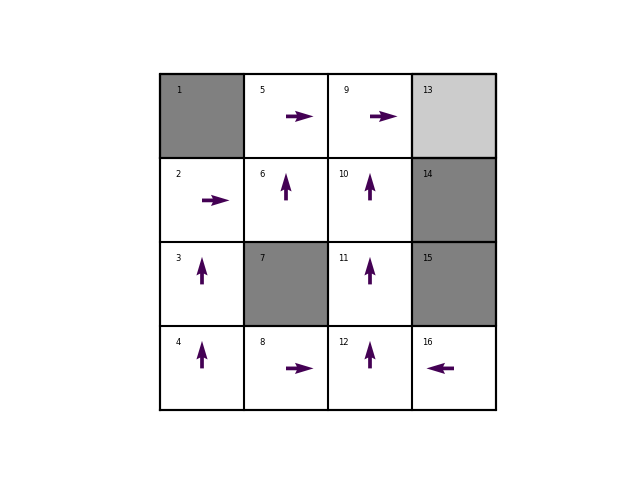
Values:



Action values:

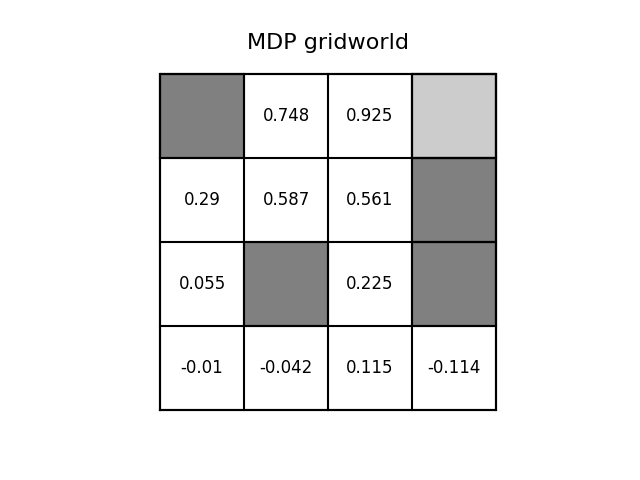


Policy:

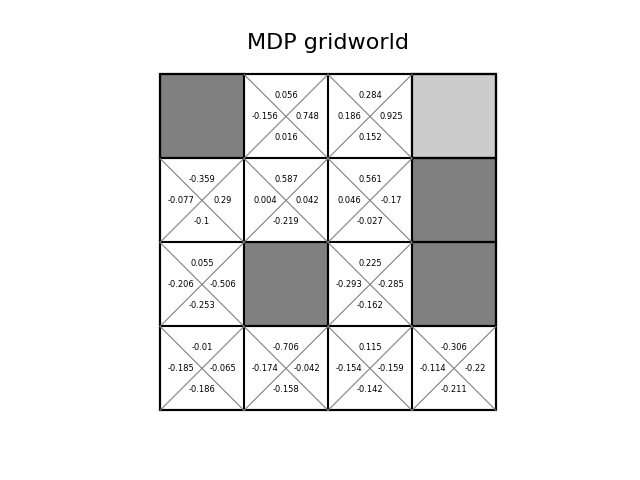


Optimal values – SARSA(after 14k episodes):

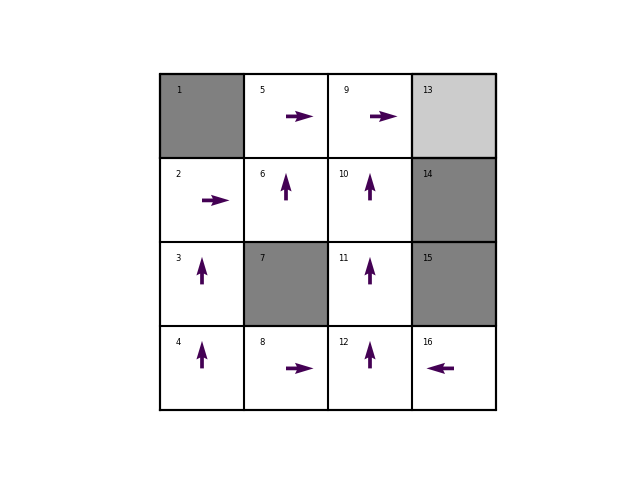
Values:



Action values:

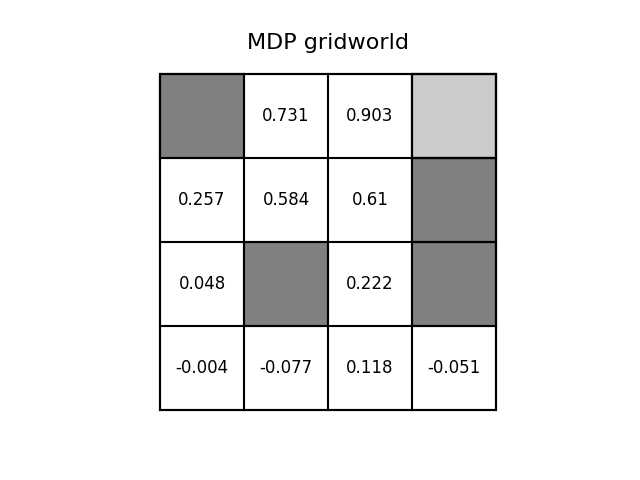


Policy:

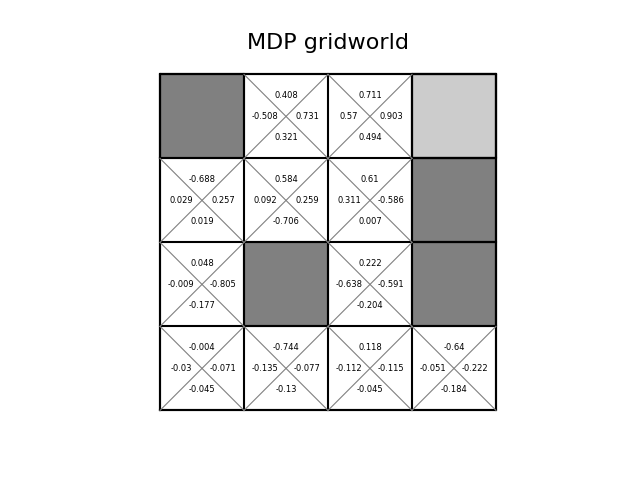


Optimal values – Qlearning(after 20k episodes):

Values:



Action values:



Policy:

