You will find the code for this question in the `RL-book/Assignment15/assignment15_pb1.py` file.

We get the following output:

```
------------- MONTE CARLO VALUE FUNCTION --------------
{'A': 9.571428571428571, 'B': 5.642857142857143}
-------------- MRP VALUE FUNCTION ----------
{'A': 12.933333333333328, 'B': 9.599999999999998}
------------- TD VALUE FUNCTION --------------
{'A': 13.252972160363933, 'B': 9.885217894397968}
------------- LSTD VALUE FUNCTION --------------
{'A': 12.933333333333334, 'B': 9.600000000000001}
```

Not all give the same value functions. We notice that LSTD Value Function and MRP value functions are the same. TD Value Function is close to these value functions, but it does not give the exact same values. As for the Monte Carlo Value Function, it is a bit more different from all the other value functions.

The reason for these differences as seen in class is that each algorithm seeks to minimize a different distant metric.

If we refer to the $\boldsymbol{\phi}$ subspace VF vectors by their weights $\boldsymbol{w}$, in Monte-Carlo with linear function approx (like it is the case in our tabular approach), we slowly converge to $\boldsymbol{w}$ that minimizes $d(\boldsymbol{V^{\pi}}, \boldsymbol{V_w})$.

In the Tabular TD(0) algorithm, we are minimizing the Bellman Error. And in the LSTD algorithm and the one with the MRP, we are minimizing the Projected Bellman Error.

This explains the similarities and differences observed.

The code for my implementation of TD with gradient correction for prediction can also be found in the `RL-book/Assignment15/assignment15_pb1.py` file. We chose to do it in the same file because we then tested our TDC Prediction algorithm against the algorithms from the previous question on the simple MRP example from that question.

This gave me the following result:

```
------------- TDC VALUE FUNCTION --------------
{'B': 9.468895890431877, 'A': 12.99421400734009}
```

The prediction for the value function using the TDC algorithm is really close to what we had with the MC or the TD algorithm.