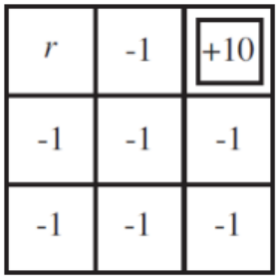
Assignment 3
Markov Decision Processes

Names	ID
Sally Kamel Soliman Armanyous	19015762
Salma Saeed Mahmoud Ghareb	19015779
Mayar Ayman Mahmoud El-khouly	19016744
Hebat-Allah Moustafa Abdel-Reheem	19016836

Problem Statement:

Consider the 3x3 world shown in the following figure:



The agent has four actions Up, Down, Right and Left.

The transition model is: 80% of the time the agent goes in the direction it selects; the rest of the time it moves at right angles to the intended direction.

Algorithms, ideas:

- **We must first identify all potential actions, stages, and transitions. Therefore, the options are as follows: right, left, up, and down.**
  There are nine states total, including the final and the "r" state.
  These are all the transitions that can be made by any state using one of the four actions.
- **The maximum reward that each state may receive must be determined using the value iteration method.**
- **As a result, we start by obtaining the transitions, which evaluates each state's possible actions and stores them in the transitions dictionary.**
- **The intended utility state for the value Iteration method is stored in a 2d array, which must be changed after each iteration. It is determined as follows, using the equation below:**

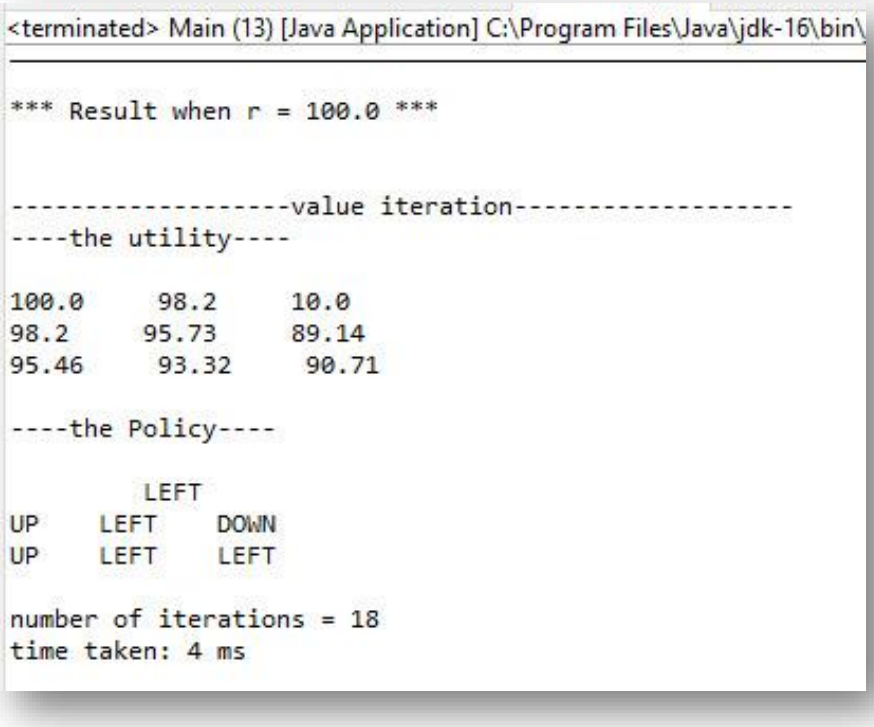
$$V_{k+1}(s) \leftarrow \max_a \sum_{s'} T(s,a,s') \left[R(s,a,s') + \gamma V_k(s') \right]$$

- **In order to obtain the new V value, we must maximize all four of the values for each state because there are four directions.**
- **Once we have completed all iterations, reached a terminal state, or if the difference between the old and new Vs is less than the set epsilon, we repeat this process until that point.**

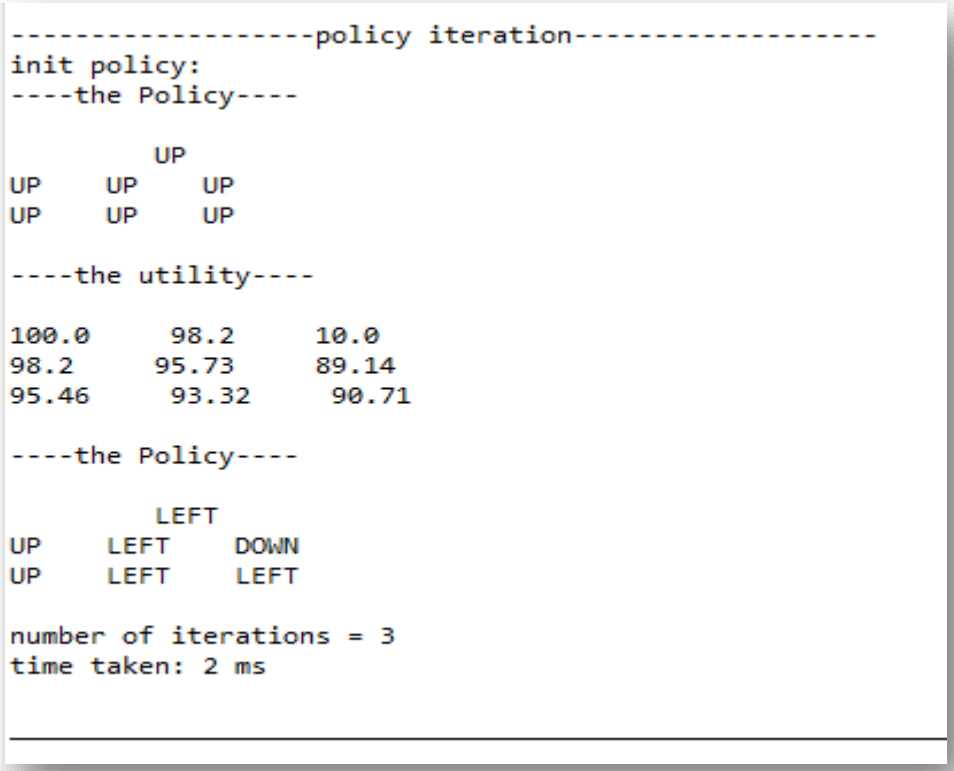
Results and required questions answers:

When r = 100 :

>> Value iteration result



>> policy iteration result



➔ **Here, the policy choses the directions to be towards the cell of +100, as in all cases its reward will be greater than any other case.**

When r = 3:

>> Value iteration result

```
<terminated> Main (13) [Java Application] C:\Program Files\Java\jdk-16\bin\

*** Result when r = 3.0 ***


--------------------value iteration--------------------
----the utility----

3.0     9.46     10.0
6.36     8.1     9.46
5.54     6.77     7.95

----the Policy----

         RIGHT
RIGHT    RIGHT    UP
RIGHT    RIGHT    UP

number of iterations = 14
time taken: 1 ms
```

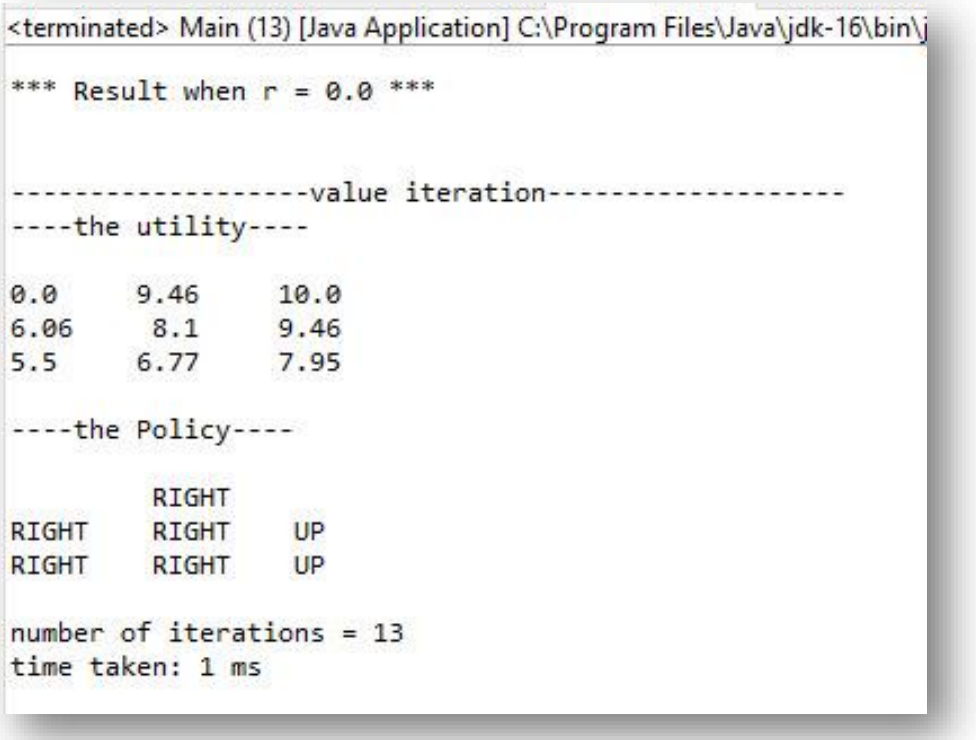
>> policy iteration result

```
--------------------policy iteration--------------------
init policy:
----the Policy----

         UP
UP    UP    UP
UP    UP    UP

----the utility----

3.0     9.46     10.0
6.36     8.1     9.46
5.54     6.77     7.95

----the Policy----

         RIGHT
RIGHT    RIGHT    UP
RIGHT    RIGHT    UP

number of iterations = 3
time taken: 0 ms
```

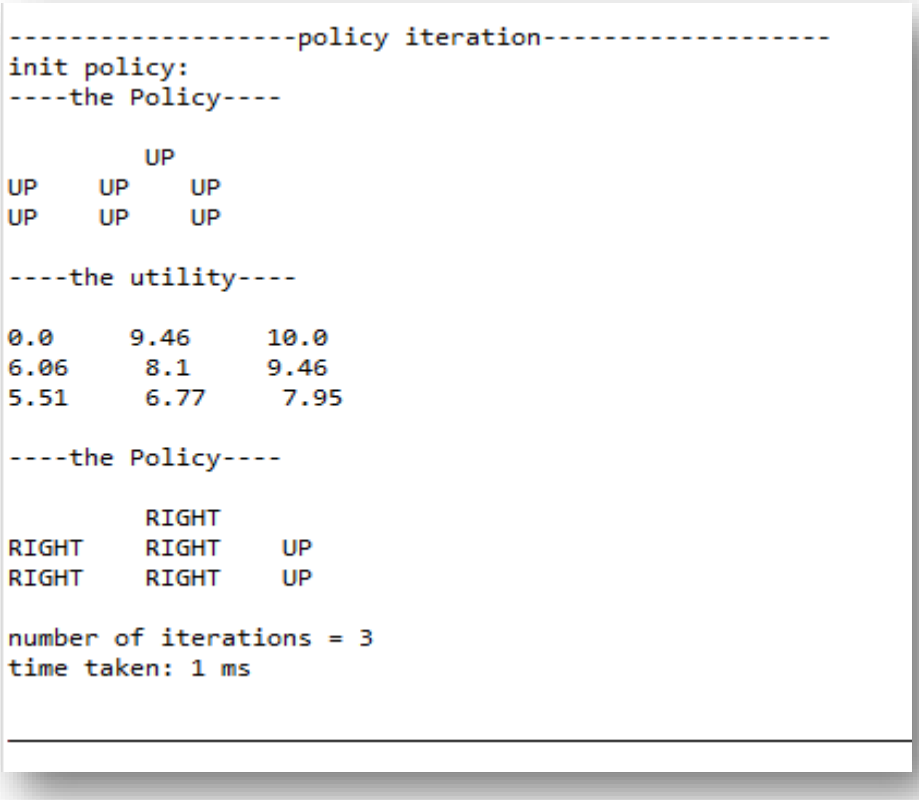
➔ **Here, the policy choses the directions to be towards the cell of +10 to collect more rewards than going through the 3 reward cell.**

When r = 0:

>> Value iteration result



>> policy iteration result



➔ **Here, the policy also chooses the directions to be towards the cell of +10 to collect more rewards than going through the 0 cell as it has zero effect on the reward.**

When r = -3:

>> Value iteration result

```
*** Result when r = -3.0 ***


--------------------value iteration--------------------
----the utility----

-3.0      9.46      10.0
5.76      8.1      9.46
5.46      6.77      7.95

----the Policy----

          RIGHT
RIGHT     RIGHT     UP
RIGHT     RIGHT     UP

number of iterations = 13
time taken: 1 ms
```

>> policy iteration result

```
-------------------policy iteration-------------------
init policy:
----the Policy----

          UP
UP     UP     UP
UP     UP     UP

----the utility----

-3.0      9.46      10.0
5.76      8.1      9.46
5.47      6.77      7.95

----the Policy----

          RIGHT
RIGHT     RIGHT     UP
RIGHT     RIGHT     UP

number of iterations = 2
time taken: 1 ms
```

➔ **Here, the policy chooses to always avoid the cell of -3 to not decrease its reward. If it goes through any other cell it will be better**