Reinforcement Learning
EX3 - Dynamic Programming

Q5 -> GridWorld 5x5

a)

```
C:\Users\ravin\Desktop\Fall'23\RL\ex3\venv\Scripts\python.exe "C:\Users\ravin\Desktop\
Value Function for iterative policy evaluation
 [-0.8 -0.5 -0.5 -0.6 -0.9 -0.5 -0.   0.1 -0.1 -0.5  0.   0.9  0.9  0.7
  0.   0.9  3.1  2.3  2.1  0.8  1.9  9.5  4.3  5.6  1.6]

Process finished with exit code 0
```

b)

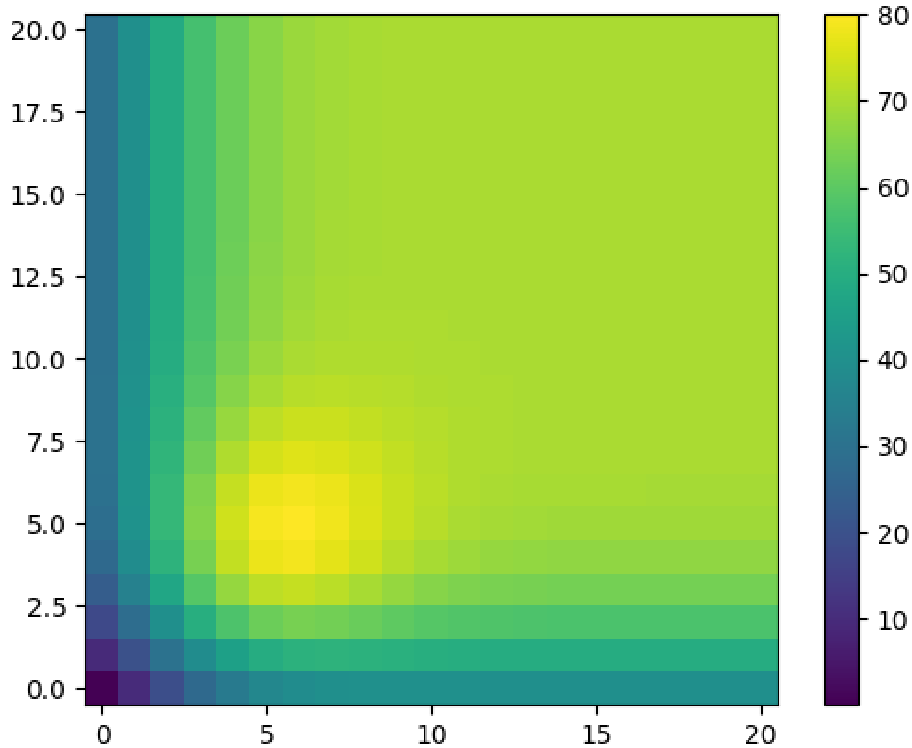
```
C:\Users\ravin\Desktop\Fall'23\RL\ex3\venv\Scripts\python.exe "C:\Users\ravin\Desktop\Fall'23\RL\ex3\venv\main.py
Optimal Value function for Value Iteration
 [14.4 16.  14.4 13.  11.7 16.  17.8 16.  14.4 13.  17.8 19.8 17.8 16.
 14.4 19.8 22.  19.8 17.8 16.  22.  24.4 22.  19.4 17.5]
('Optimal policy for Value Iteration \n'
 ' defaultdict(<function GridWorld.value_iteration.<locals>.<lambda> at '
 "0x0000011BACF4D940>, {(0, 0): ['RIGHT', 'UP'], (0, 1): ['RIGHT'], (0, 2): "
 "['DOWN', 'RIGHT'], (0, 3): ['DOWN', 'RIGHT'], (0, 4): ['DOWN', 'RIGHT'], (1, "
 "0): ['RIGHT', 'UP'], (1, 1): ['RIGHT'], (1, 2): ['DOWN', 'RIGHT'], (1, 3): "
 "['DOWN', 'RIGHT'], (1, 4): ['DOWN', 'RIGHT'], (2, 0): ['RIGHT', 'UP'], (2, "
 "1): ['RIGHT'], (2, 2): ['DOWN', 'RIGHT'], (2, 3): ['DOWN', 'RIGHT'], (2, 4): "
 "['DOWN', 'RIGHT'], (3, 0): ['RIGHT', 'UP'], (3, 1): ['RIGHT'], (3, 2): "
 "['DOWN', 'RIGHT'], (3, 3): ['DOWN'], (3, 4): ['DOWN'], (4, 0): ['UP'], (4, "
 "1): ['LEFT', 'DOWN', 'RIGHT', 'UP'], (4, 2): ['DOWN'], (4, 3): ['LEFT', "
 "'DOWN', 'RIGHT', 'UP'], (4, 4): ['DOWN']})")

Process finished with exit code 0
```

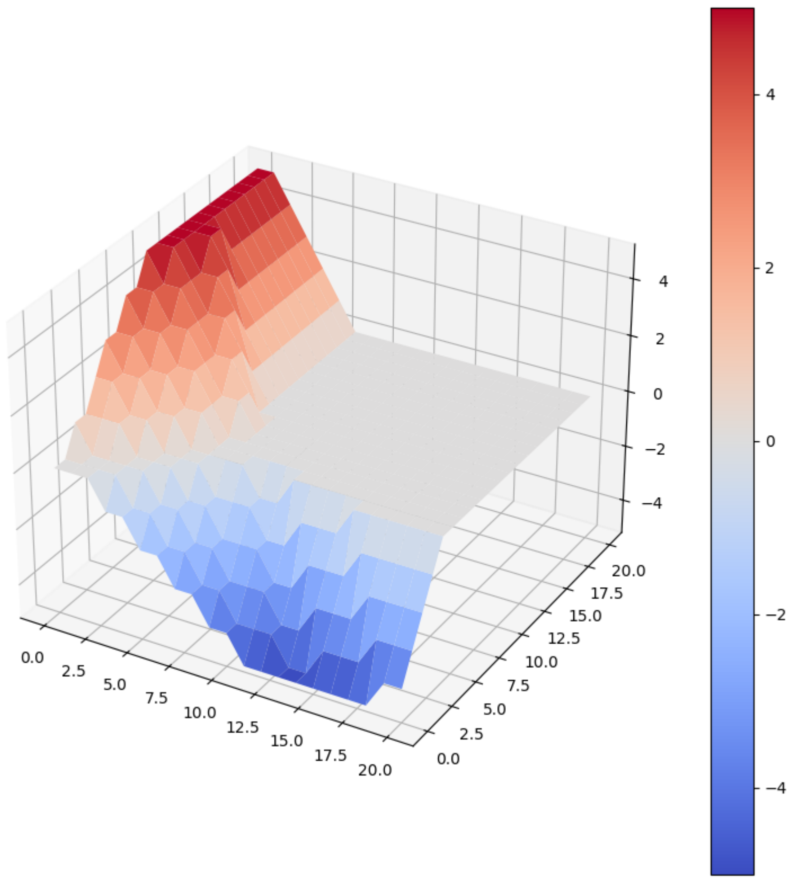
c)

```
C:\Users\ravin\Desktop\Fall'23\RL\ex3\venv\Scripts\python.exe "C:\Users\ravin\Desktop\Fall'23\RL\ex3\v
Optimal Value function for Policy Iteration
 [14.4 16.  14.4 13.  11.7 16.  17.8 16.  14.4 13.  17.8 19.8 17.8 16.
 14.4 19.8 22.  19.8 17.8 16.  22.  24.4 22.  19.4 17.5]
Optimal policy function for Policy Iteration
 [[0.   0.   0.5  0.5 ]   L, D, R, U
 [0.   0.   1.   0.  ]
 [0.   0.5  0.5  0.  ]
 [0.   0.5  0.5  0.  ]
 [0.   0.5  0.5  0.  ]
 [0.   0.   0.5  0.5 ]
 [0.   0.   1.   0.  ]
 [0.   0.5  0.5  0.  ]
 [0.   0.5  0.5  0.  ]
 [0.   0.5  0.5  0.  ]
 [0.   0.   0.5  0.5 ]
 [0.   0.   1.   0.  ]
 [0.   0.5  0.5  0.  ]
 [0.   0.5  0.5  0.  ]
 [0.   0.5  0.5  0.  ]
 [0.   0.   0.5  0.5 ]
 [0.   0.   1.   0.  ]
 [0.   0.5  0.5  0.  ]
 [0.   1.   0.   0.  ]
 [0.   1.   0.   0.  ]
 [0.   0.   0.   1.  ]
 [0.25 0.25 0.25 0.25]
 [0.   1.   0.   0.  ]
 [0.25 0.25 0.25 0.25]
 [0.   1.   0.   0.  ]]
```

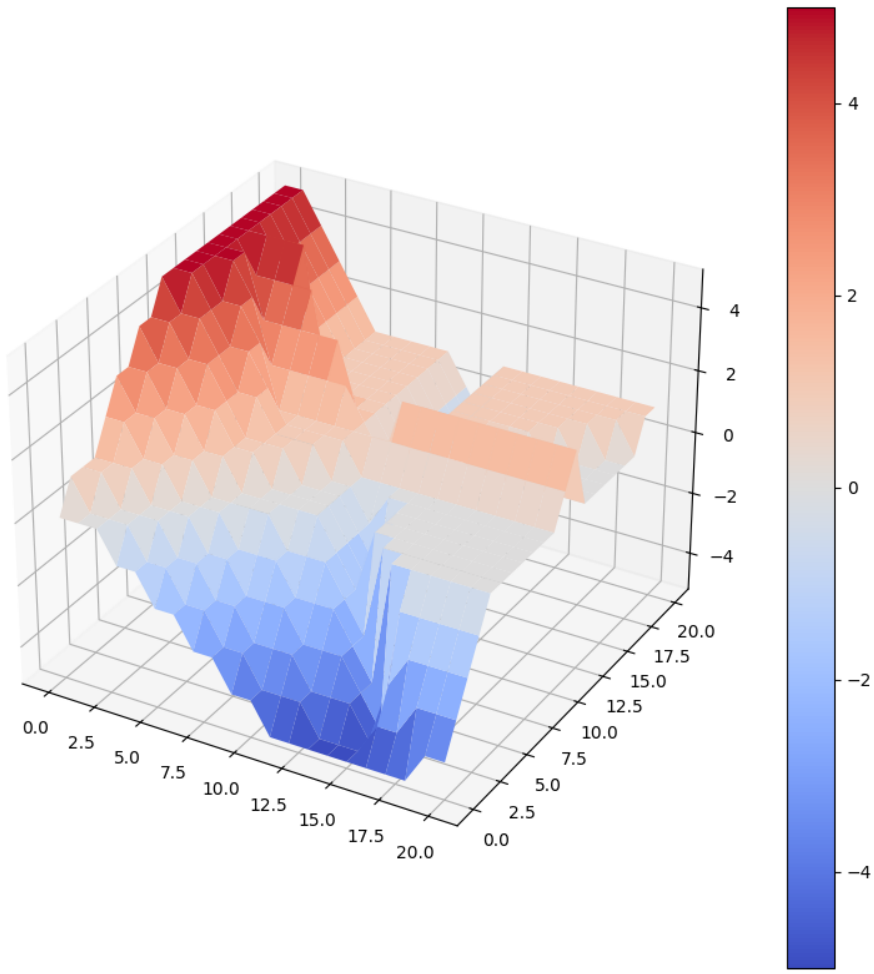
Q6)
a)Optimal Policy



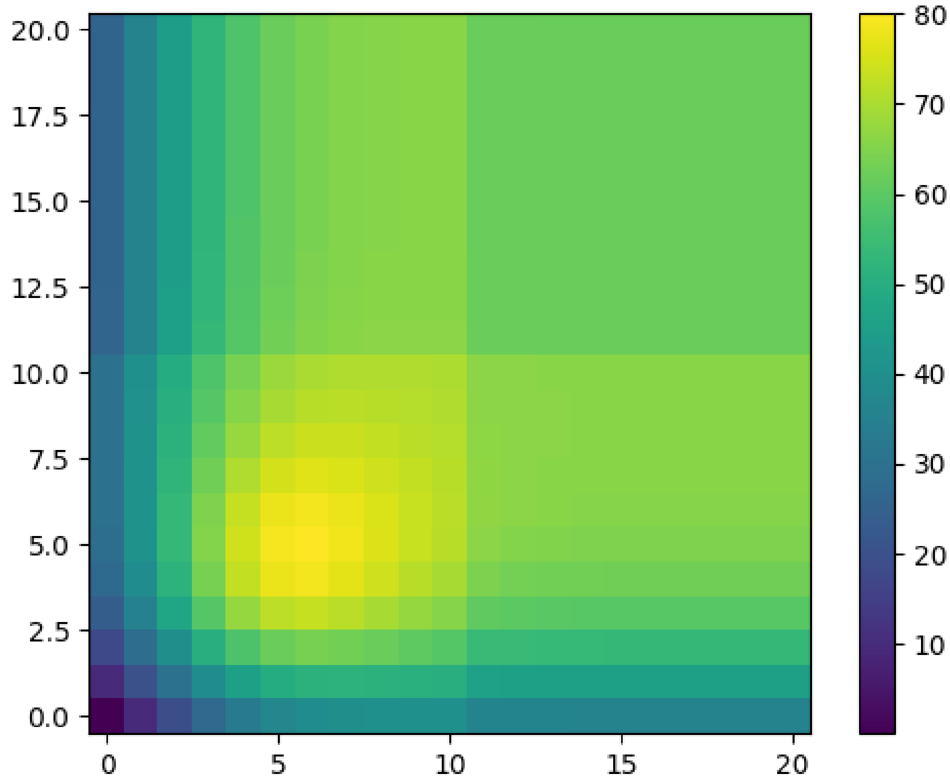
Optimal value function



b) Modified Jacks Car Rental →
Optimal value Function →



Optimal policy Function →



written: How do we change dynamics function to reflect modified Jack's car rental problem ?

1. The cost of moving a car from one location to another is reduced by $2 if the number of cars being moved is positive. as the rental company is able to take advantage of a free car move.
2. Plus there is additional cost of 4 dollars for storing and managing a large number of cars at a single location
3. Will improve the way we calculate reward to ensure we also consider the cases with changed costs.

Written: How does the final policy differ from the original Jack's Car Rental Problem ?

The optimal policy tries to move at least 1 car from location A to location B since
it's free to move But, when location B has exactly 10 cars, since the rental car company needs
to pay overflow parking costs.