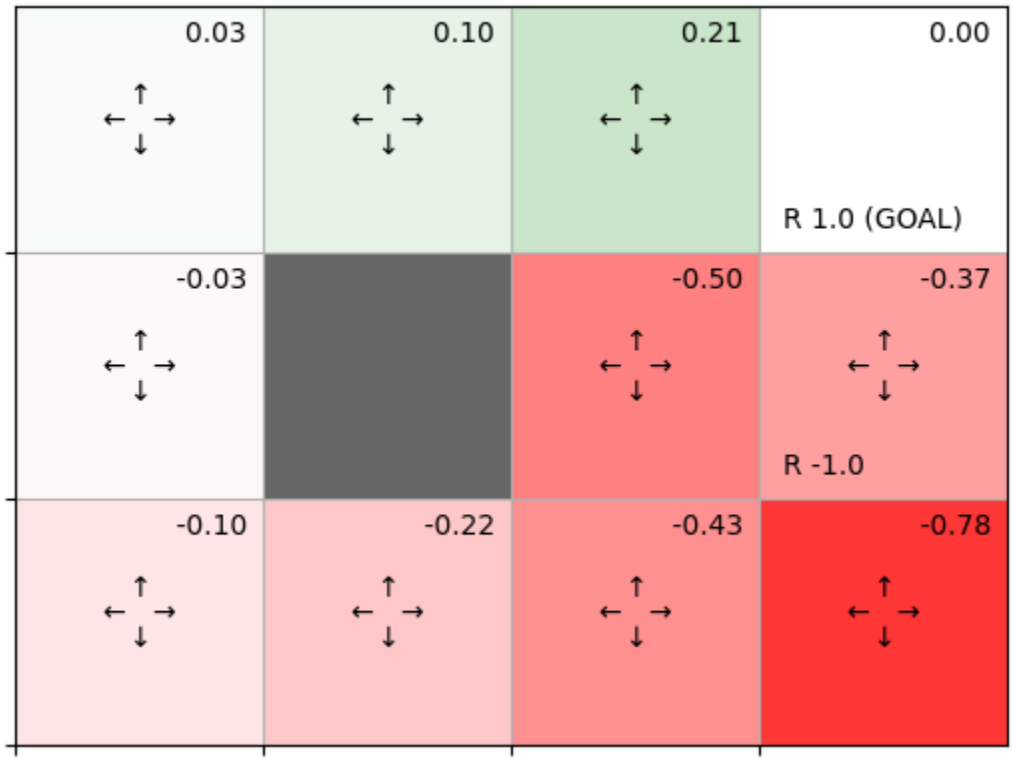
실습1

코드 ->

```
from collections import defaultdict
from gridworld import GridWorld

def eval_onestep(pi, V, env, gamma = 0.9):
    for state in env.states():
        if state == env.goal_state:
            V[state] = 0
            continue

        action_probs = pi[state]
        new_V = 0

        for action,action_prob in action_probs.items():
            next_state = env.next_state(state, action)
            r = env.reward(state, action, next_state)

            new_V += action_prob * ( r+ gamma * V[next_state])

        V[state] = new_V
    return V

def policy_eval(pi, V, env, gamma, threshold = 0.001):
    while True:
        old_V = V.copy()
        V = eval_onestep(pi, V, env, gamma)

        delta = 0
        for state in V.keys():
            t = abs(V[state] - old_V[state])
            if delta < t:
                delta = t

        if delta < threshold:
            break
    return V

if __name__ == '__main__':
    env = GridWorld()
    gamma = 0.9

    pi = defaultdict(lambda: {0: 0.25, 1:0.25, 2:0.25, 3:0.25})
    V = defaultdict(lambda: 0)

    V = policy_eval(pi, V, env, gamma)

    env.render_v(V, pi)
```

결과 :

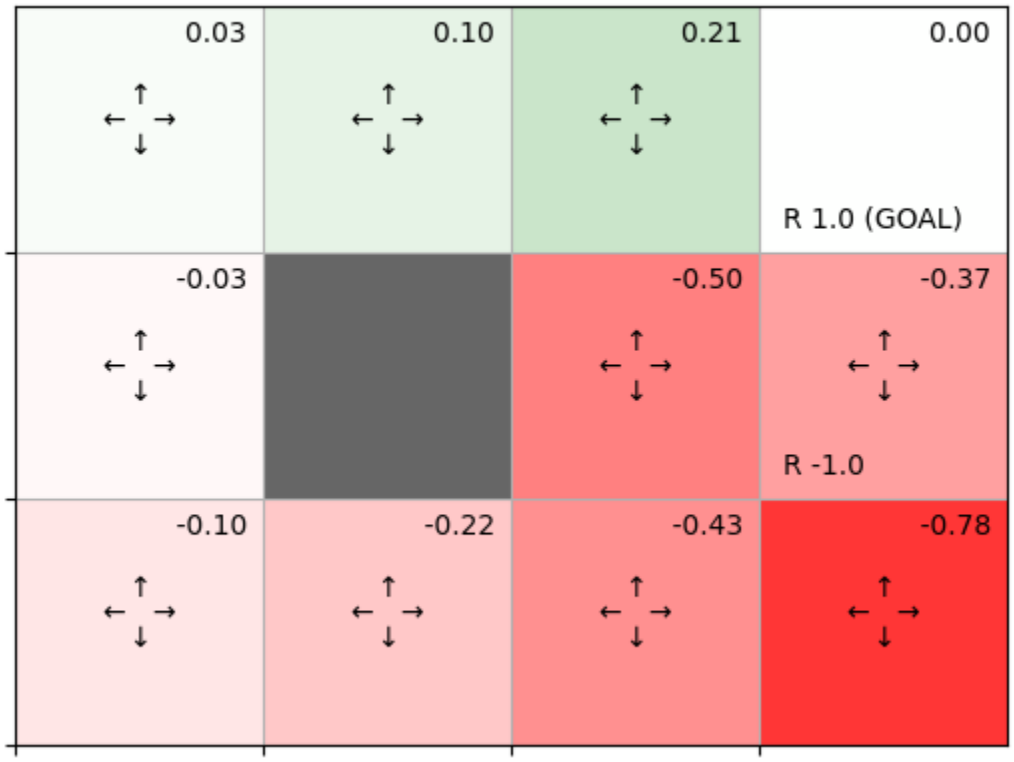

0.03
0.10
0.21
0.00
R 1.0 (GOAL)
-0.03
-0.50
-0.37
R -1.0
-0.10
-0.22
-0.43
-0.78

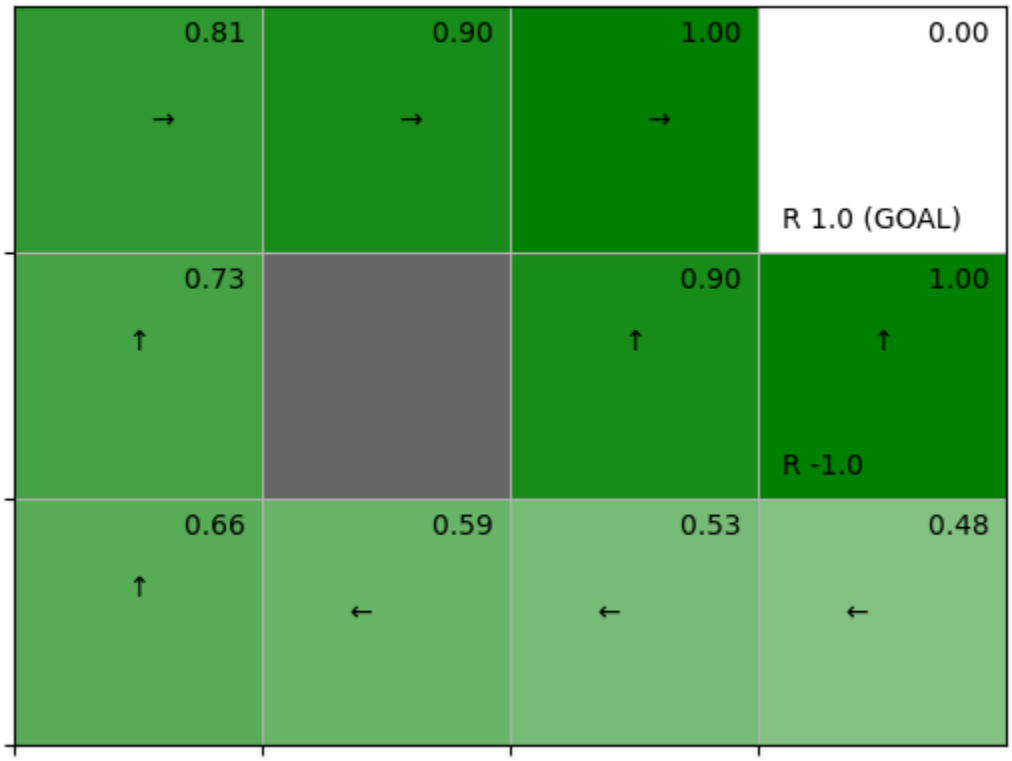
실습 2

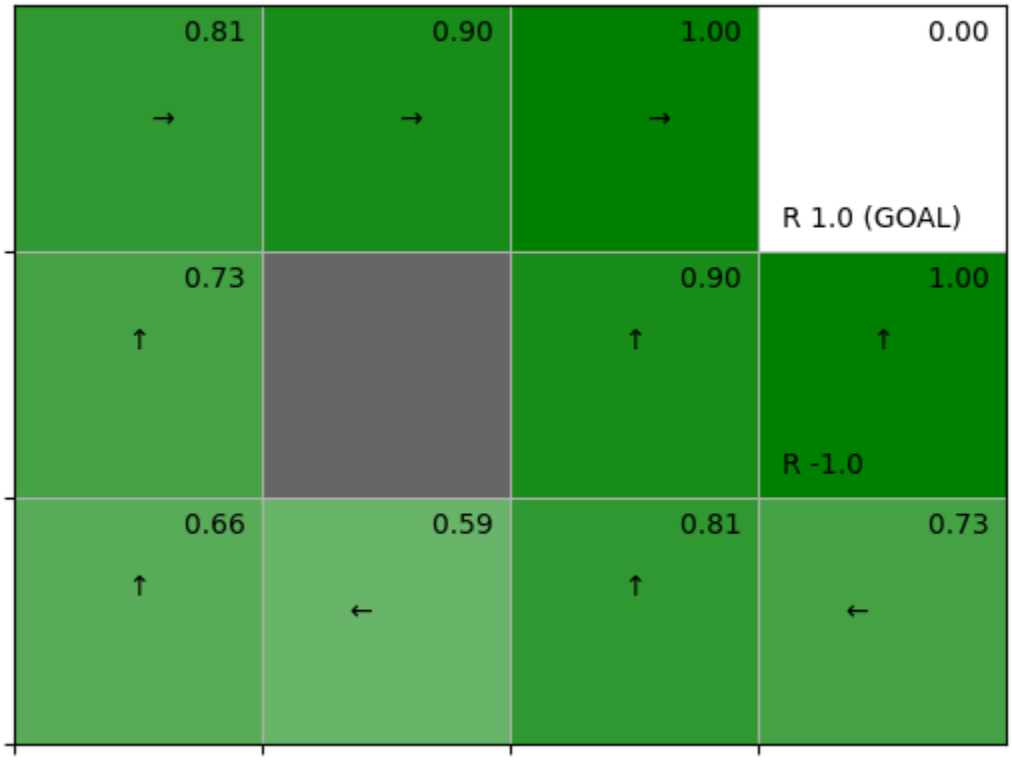
코드 ->

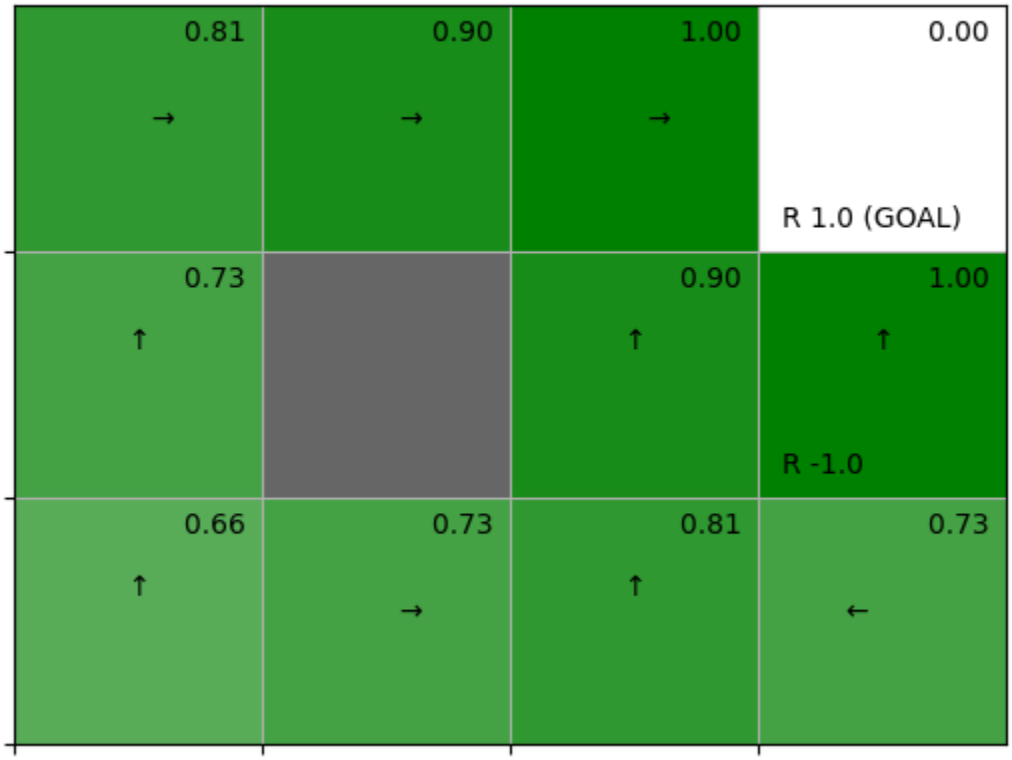
```
from collections import defaultdict
from gridworld import GridWorld
from policyEval import policyEval


def argmax(d):
    """d (dict)"""
    max_value = max(d.values())
    max_key = -1
    for key, value in d.items():
        if value == max_value:
            max_key= key


    return max_key

def greedy_policy(V, env, gamma):
    pi = {}

    for state in env.states():
        action_values = {}

        for action in env.actions():
            next_state = env.next_state(state, action)
            r = env.reward(state, action, next_state)
            value = r + gamma * V[next_state]
            action_values[action] = value

        max_action = argmax(action_values)
        action_probs = {0: 0, 1: 0, 2: 0, 3:0}
        action_probs[max_action] = 1.0
        pi[state] = action_probs
    return pi

def policy_iter(env, gamma, threshold=0.001, is_render = True):
    pi = defaultdict(lambda: {0: 0.25, 1: 0.25, 2: 0.25, 3: 0.25})
    V = defaultdict(lambda: 0)

    while True:
        V = policyEval(pi, V, env, gamma, threshold)
        new_pi = greedy_policy(V, env, gamma)

        if is_render:
            env.render_v(V, pi)

        if new_pi == pi:
            break
        pi = new_pi
    return pi



if __name__ == '__main__':
    env = GridWorld()
    gamma = 0.9

    pi = policy_iter(env, gamma)
```

결과 :


0.03
0.10
0.21
0.00
R 1.0 (GOAL)
-0.03
-0.50
-0.37
R -1.0
-0.10
-0.22
-0.43
-0.78


0.81
0.90
1.00
0.00
R 1.0 (GOAL)
0.73
0.90
1.00
R -1.0
0.66
0.59
0.53
0.48


0.81
0.90
1.00
0.00
R 1.0 (GOAL)
0.73
0.90
1.00
R -1.0
0.66
0.59
0.81
0.73

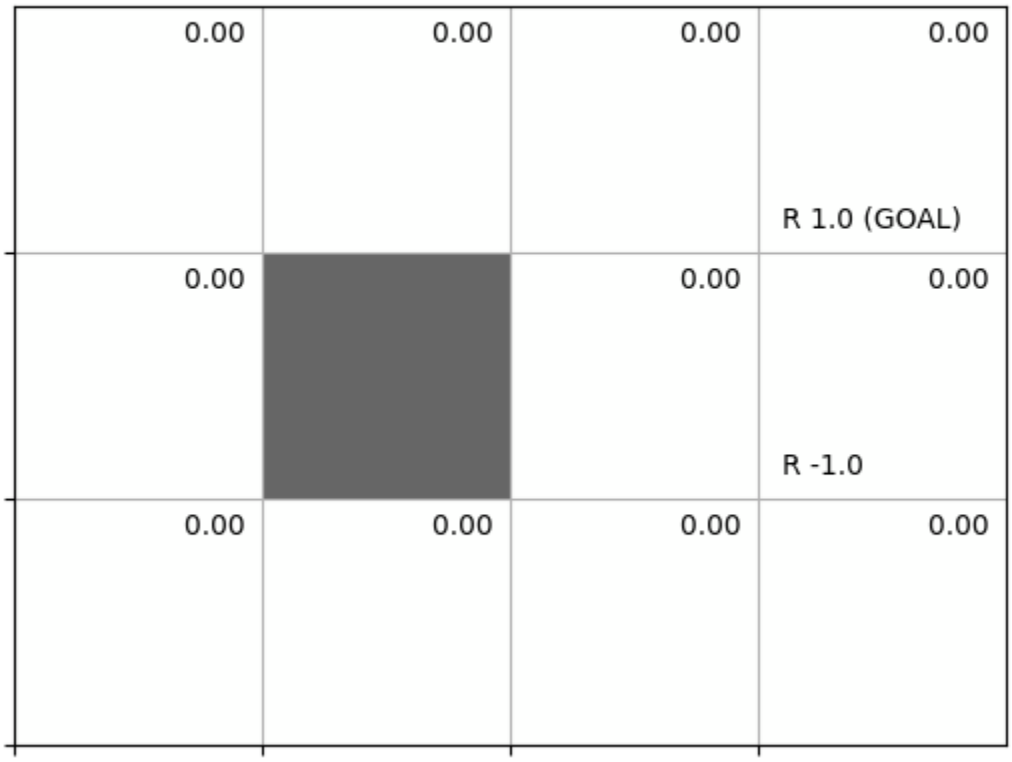

0.81
0.90
1.00
0.00
R 1.0 (GOAL)
0.73
0.90
1.00
R -1.0
0.66
0.73
0.81
0.73

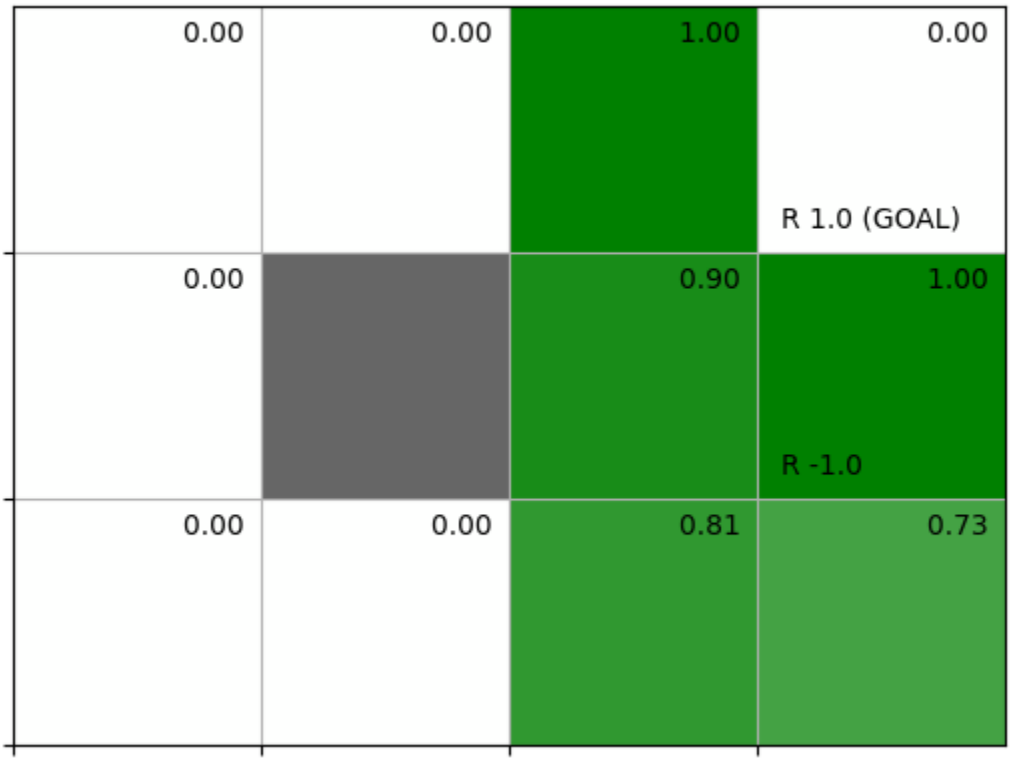
실습 3 :

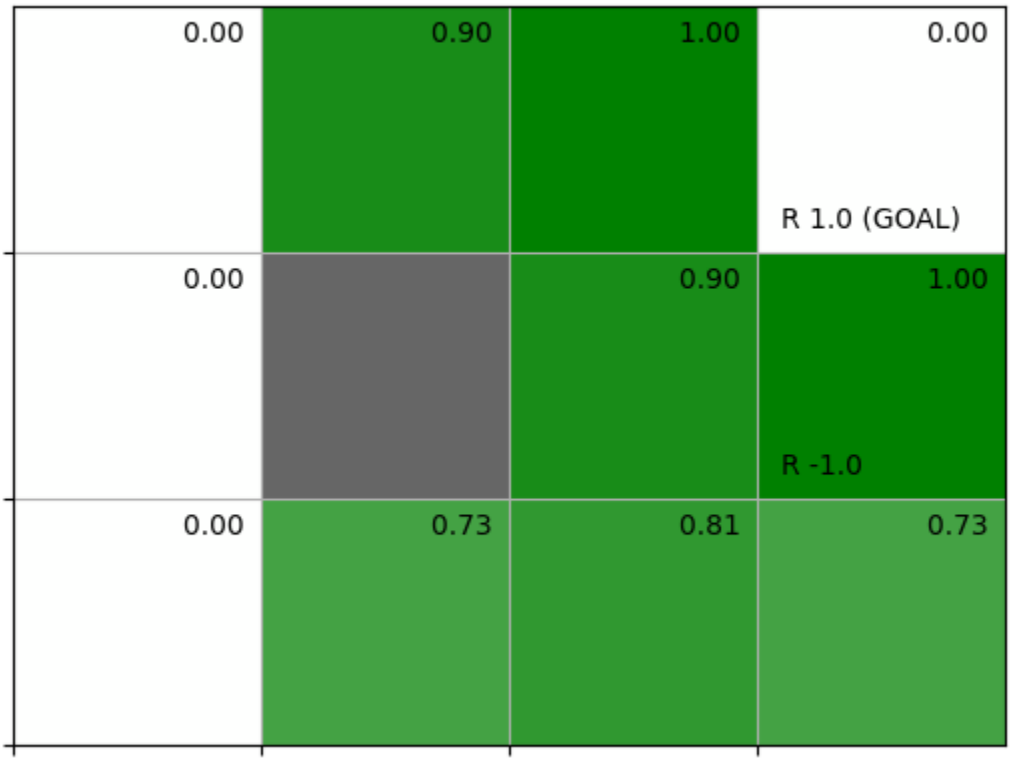
코드

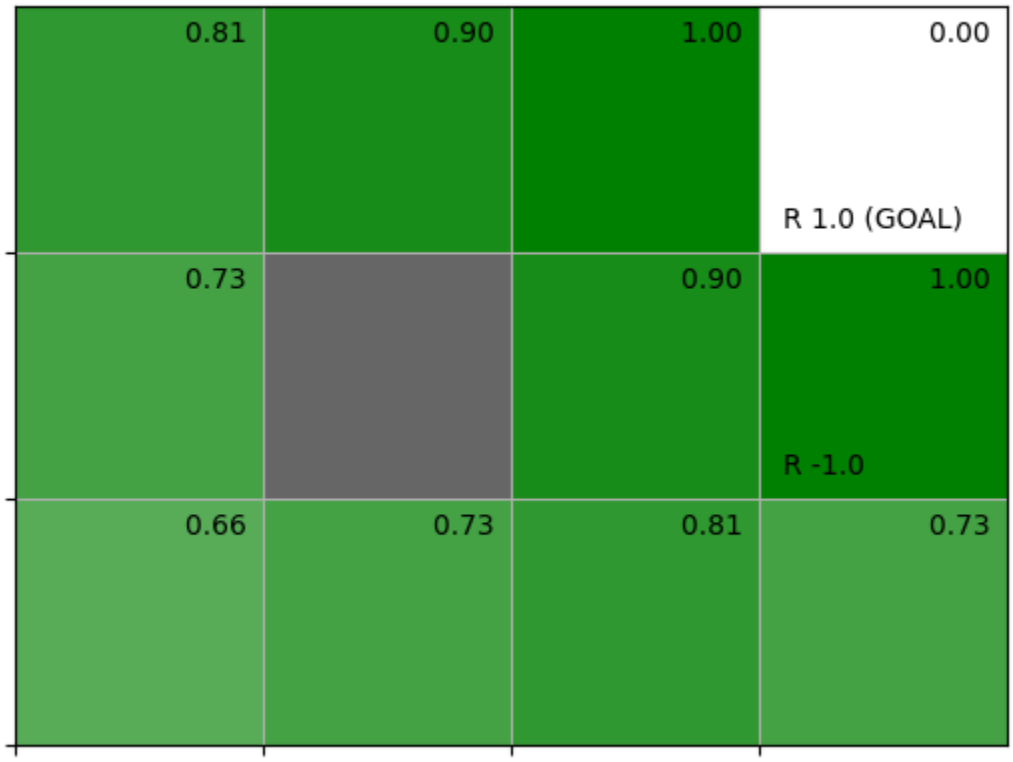
```python
from collections import defaultdict
from gridworld import GridWorld
from policyIter import greedy_policy


def value_iter_onestep(V, env, gamma):
    for state in env.states():
        if state == env.goal_state:
            V[state] = 0
            continue

        action_values = []
        for action in env.actions():
            next_state = env.next_state(state, action)
            r = env.reward(state, action, next_state)
            value = r + gamma * V[next_state]
            action_values.append(value)

        V[state] = max(action_values)
    return V

def value_iter(V, env, gamma, threshold = 0.001, is_render = True):
    while True:
        if is_render:
            env.render_v(V)
        old_V = V.copy()
        V = value_iter_onestep(V, env, gamma)

        delta = 0

        for state in V.keys():
            t = abs(V[state] - old_V[state])
            if delta < t:
                delta = t

        if delta < threshold:
            break
    return V



if __name__ == '__main__':
    V = defaultdict(lambda: 0)
    env = GridWorld()
    gamma = 0.9

    V = value_iter(V, env, gamma)

    pi = policy_iter(env, gamma)
    env.render_v(V, pi)
```

결과




0.00
0.00
1.00
0.00
R 1.0 (GOAL)
0.00
0.90
1.00
R -1.0
0.00
0.00
0.81
0.73


0.00
0.90
1.00
0.00
R 1.0 (GOAL)
0.00
0.90
1.00
R -1.0
0.00
0.73
0.81
0.73


0.81
0.90
1.00
0.00
R 1.0 (GOAL)
0.73
0.90
1.00
R -1.0
0.66
0.73
0.81
0.73