EXPERIMENT - 10

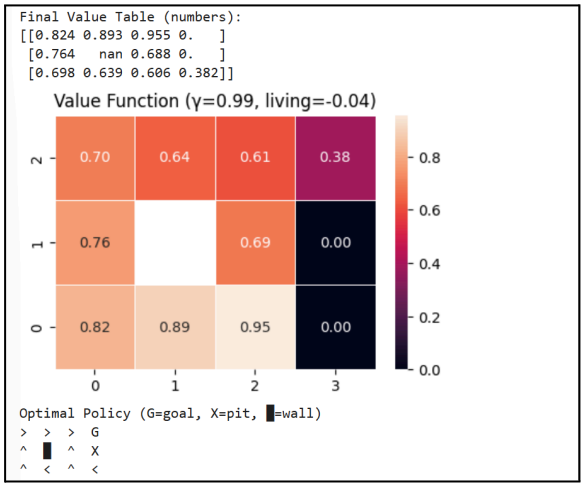
Project Report (23/CS/062).

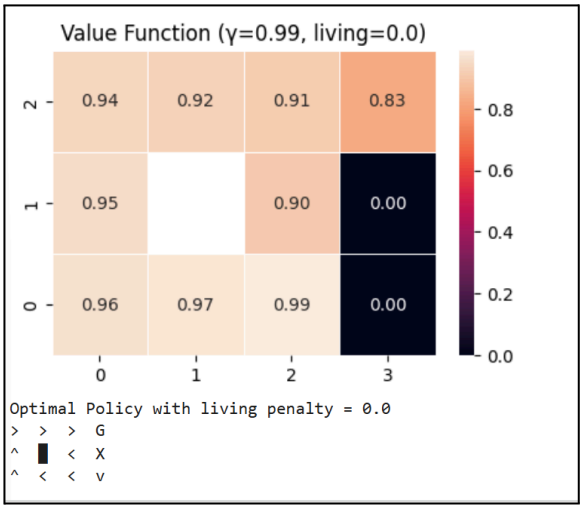
Code Snippets. (Value_iteration , Policy_extraction).

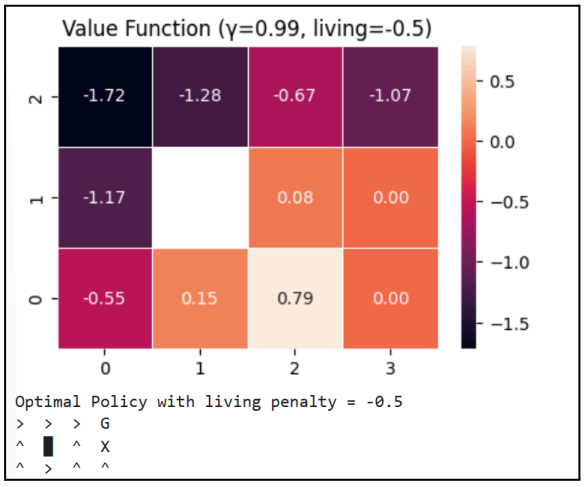
```
# Cell 3: Value Iteration
def q_value(s, a, V, R, gamma):
    """Q(s,a) = sum_p [ p * ( R(ns) + gamma * V[ns] ) ]"""
    return sum(p * (R[ns] + gamma * V[ns]) for p, ns in get_next_states(s, a))

def value_iteration(gamma=0.99, theta=1e-4, living_penalty=-0.04,
verbose=False):
    S = states_list()
    R = make_reward(living_penalty)
    V = {s: 0.0 for s in S}

    # Terminals often set to 0 for post-entry value; their immediate rewards
are in R
    V[GOAL] = 0.0
    V[PIT]  = 0.0

    while True:
        delta = 0.0
        for s in S:
            if is_terminal(s):
                continue
            q_vals = [q_value(s, a, V, R, gamma) for a in ACTIONS]
            v_new = max(q_vals)
            delta = max(delta, abs(v_new - V[s]))
            V[s] = v_new
        if verbose:
            print(f"delta = {delta:.6f}")
        if delta < theta:
            break
    return V, R



# Cell 4: Extract optimal policy from V
ARROWS = {"up":"^", "down":"v", "left":"<", "right":">"}

def extract_policy(V, R, gamma=0.99):
    Pi = {}
    for s in states_list():
        if s == GOAL:
```

```
            Pi[s] = "G"
        elif s == PIT:
            Pi[s] = "X"
        elif s == WALL:
            continue
        else:
            qs = {a: q_value(s, a, V, R, gamma) for a in ACTIONS}
            best_a = max(qs, key=qs.get)
            Pi[s] = best_a
    return Pi
```

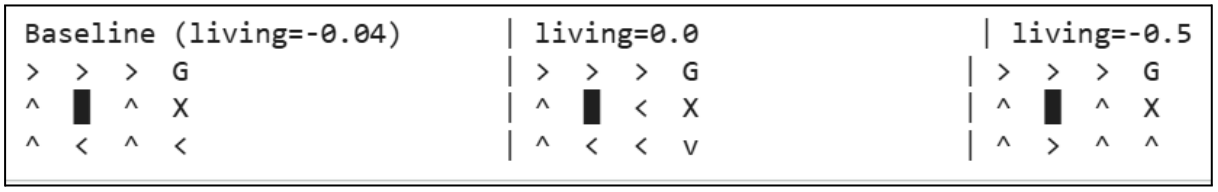
Final Results.







Policy Comparison.



ANALSIS.

The baseline policy with **γ = 0.99** and **living penalty = −0.04** successfully guides the agent toward the goal while avoiding the pit. The value table shows a clear gradient toward higher values near the goal (≈0.95), and the policy arrows indicate a **safe, slightly longer route**, reflecting a cautious strategy encouraged by the small living penalty.

When the **living penalty was reduced to 0.0**, the agent's incentive to minimize steps decreased, leading to higher state values since no penalty is incurred per move. However, the **policy remained almost unchanged**, as the dominant terminal rewards (goal = +1, pit = −1) still determined the optimal actions.

With a **higher penalty (−0.5)**, the agent's behavior shifted notably—values became negative, and the policy favored **shorter but riskier paths**. The strong penalty per step made efficiency more important than safety, pushing the agent to reach the goal faster even if it passed near the pit.

Overall, the results show that the **living penalty directly influences risk-taking and path length**:

- Small penalty → safe, longer path
- No penalty → neutral behavior
- High penalty → fast, riskier path

The provided **value and policy tables** are sufficient for analysis, as they clearly illustrate how penalty adjustments affect the agent's decision-making.