90

Dakota Walker
CSC449
Exam 2

1)
The initial learning rate is well north of 1 when using alpha = $1/\log_{10}(n+1)$ and a minor remaining trace amount for n. Previous states with decayed traces will have greater learning rates than the more recent traced states. I changed the bounds of alpha to (0,1] to help keep the step sizes down.

When choosing the initial action, the faulty code uses a uniform distribution over the actions rather than following the policy. While the uniform distribution method may work for the very first episode, the following episodes should follow the policy or something like e-greedy.

For the internal **if** statement checking for terminal state, this should be done after the current state is updated to the next state. That way you can eliminate the code in the middle that offers two different future reward values to be used in the Q table update.

When updating v_t in the else statement, the future action chosen by the policy should be used for SARSA. The discount factor for the future rewards was also missing.

Q update function updates the wrong spot in the table.

Traces are accumulated, but never decayed. All traces need to be reset between episodes. Renamed trace vector from *n* to *z* for readability and added a decay variable γ, even if the assumption is that it was equal to 1 and therefore not needed.

```
CORRECTED procedure SARSA(
num episodes $n \in N$
discount factor $\lambda \in (0, 1]$
trace decay factor $\gamma \in (0, 1]$
learning rate $\alpha_n \in (0, 1]$
Initialize matrices Q(s, a) and z(s, a) to 0, $\forall$(s, a)
for episode $k \in 1, 2, 3, \ldots, n$
    $t \leftarrow 1$
    Initialize $s_1$
    Choose $a_1$ from $s_1$ using $\mu_t$ : an ε-greedy policy with respect to Q
    z(s, a) to 0, $\forall$(s, a)
    while Episode k is not finished
        Take action $a_t$ : observe reward $r_t$ and next state $s_{t+1}$
        Choose $a_{t+1}$ from $s_{t+1}$ using $\mu_t$ : an ε-greedy policy with respect to Q
        $y = r_t + \lambda \max_a(Q(s_{t+1}, a_{t+1}))$    Compute Target Value
        $z(s_t, a_t) \leftarrow z(s_t, a_t) + 1$    Accumulating Traces
        Update Q table and traces functions for all (s,a):
            $Q(s, a) \mathrel{+}= \alpha_{z(s,a)}(y - Q(s_t, a_t))$
            $z(s, a) \leftarrow \lambda\gamma z(s, a)$
        $s_t \leftarrow s_{t+1}$
        $a_t \leftarrow a_{t+1}$
        if the current state is terminal
            episode = finished
        end if
    end while
end for
end procedure
```

no max

2.)

a) Policies with deterministic state transitions and controlled exploration should be used for Expected SARSA to stay on-policy.

example?

b) If the variant of Expected SARSA is off-policy, then it could behave similarly to Q-Learning with its exploring and learning off policy. The Q values would eventually converge in that case, as long as the terminal state offered a reward (Q value) that countered the penalties (>= 0 in this case) to propagate through even if the base policy never changes.