# Homework-1 Problem 2

**Pseudo Code:**

1. Initialize all residents as unmatched.

2. While there are unmatched residents:

a. The resident proposes to their highest-ranked hospital that has not yet rejected them.

b. The hospital tentatively accepts residents based on preference, up to its quota.

c. If the hospital exceeds its quota, it rejects the least preferred resident.

d. The rejected resident moves to the next hospital in their preference list.

3. Repeat until no resident remains unmatched.

4. Return the final stable matching.

**Description of Pseudo code:**

The algorithm follows an iterative approach where each resident proposes to hospitals based on their preference list.

- A queue is used to track unmatched residents, ensuring all have a chance to be assigned.

- Hospitals tentatively accept residents up to their quota, following their own preference lists.

- If a hospital is over capacity, it removes the least preferred resident, who then re-enters the unmatched pool.

- This continues until no unmatched residents remain, ensuring a stable matching.

**Proof of Correctness:**

**Termination:** The algorithm will always terminate because each resident can only propose to each hospital once. Given there are r residents and h hospitals, the total number of proposals is at most r × h, ensuring termination.

Stability:

**1. No Resident Trades Down:** A resident only switches hospitals if they are rejected by a higher-preference hospital. Thus, no resident will break a match to opt for a lower-preferred hospital.

**2. No Unmatched Resident/Hospital Pair:** All hospitals fill their available slots if possible, and residents are either matched or left unmatched due to the constraints of hospital quotas.

**3. No Unstable Pair Exists**: If a hospital-resident pair (h, r) is not matched, it means either the resident prefers their current hospital over h, or the hospital prefers its current residents over r. This ensures that no hospital-resident pair has an incentive to deviate from the assigned match, guaranteeing stability.