The assignment problem requires assigning `n` different topics to `n` students based on their preferences, ensuring each student receives exactly one topic they like. To solve this, we employed a dynamic programming approach using bitmasks. The state is defined as `dp[mask]`, where `mask` is a bitmask representing the topics already assigned. If the `i-th` bit in `mask` is set, the `i-th` topic has been assigned. The value of `dp[mask]` denotes the number of ways to assign topics for the given `mask`.

The base case is `dp[(1 << n) - 1] = 1`, indicating there's one way to assign topics when all topics are assigned. The transition involves iterating over all masks and counting the number of set bits to determine which student's preferences to consider. For each student and topic, if the student likes the topic and it is not assigned in `mask`, we update `dp[mask]` based on possible assignments. The final result, stored in `dp[0]`, represents the number of ways to assign all topics starting from an empty assignment.The implementation in C involves initializing the preference matrix, setting up the DP array, and performing a bottom-up calculation to fill the DP table based on student preferences and current assignments.