

1. **Reading Input:** The program reads the number of students  $n$  and then the  $n \times n$  preference matrix.
2. **Dynamic Programming Array:**  $dp[mask]$  represents the number of ways to assign topics to the set of students represented by mask. Here, mask is a bitmask where the  $i$ -th bit is set if the  $i$ -th topic is assigned.
3. **Initialization:**  $dp[0] = 1$  because there is exactly one way to assign topics when no topics are assigned.
4. **Filling DP Array:** For each mask, it calculates the number of assignments by iterating over each topic. If the topic is part of the current mask and the current student (determined by the number of bits set in mask) likes that topic, it adds the number of assignments possible without that topic.
5. **Result:** The result is stored in  $dp[(1 \ll n) - 1]$ , which represents the number of ways to assign all topics to all students.

This approach efficiently counts the number of valid assignments using bit manipulation and dynamic programming.