- The C program addresses the problem of assigning topics to students based on their preferences.
- Each student must get exactly one preferred topic.
- Key Components:
  - `preferences` matrix: Denotes student-topic preferences (`preferences[i][j] = 1` if student `i` likes topic `j`).
  - `dp` array: Used for memoization to store the number of valid assignments for a given state.
  - Recursive function `countWays(int mask)`: Uses memoization to count valid assignments.
- Input Handling:
  - Reads the number of test cases `t`.
  - For each test case, reads `n` (number of students/topics) and populates the `preferences` matrix.

-Initialization:

  - Initializes the `dp` array to `-1` to indicate unsolved subproblems.
- Bitmask `mask`:
  - Represents the assignment state where each bit indicates if a topic is assigned.
- Recursive Function:
  - Counts valid assignments by considering unassigned topics liked by the current student.
  - Uses `__builtin_popcount(mask)` to determine the number of assigned students.
  - If all students are assigned, returns `1`.
  - Recursively explores all possible assignments and uses the `dp` array to avoid redundant calculations.
- Example:
  - For `n = 3` and a specific `preferences` matrix, the program explores all valid assignments recursively.
  - Ensures an optimal solution by leveraging dynamic programming and bitmasking.
- Efficiency:
  - The program efficiently handles the exponential number of possible assignments by using memoization to avoid redundant calculations.