This C program solves the problem of assigning topics to students based on their preferences. The program reads the number of students, their preferences, and outputs the number of different assignments of topics to students such that each student gets exactly one topic they like.

Variables and Data Structures

'Pref': a 2D array of size 20x20 to store the preferences of each student. Pref [i] [j] is 1 if student i likes topic j, and 0 otherwise.

assigned: an array of size 20 to keep track of which topic is assigned to each student. Assigned [i] stores the topic assigned to student i, or -1 if no topic is assigned.

n: the number of students.

Functions 'Is' Assigned (topic): checks if a topic is already assigned to a student. Returns 1 if the topic is assigned, and 0 otherwise.

Assign Topics (student): a recursive function that tries to assign topics to students. Returns the number of valid assignments found.

main(): the entry point of the program, responsible for reading input, initializing variables, and calling the assign Topics function.

Algorithm : The program reads the number of students n and their preferences, storing them in the pref array.

The assign Topics function is called with the first student (0) as an argument.

The function checks if the current student is the last student (student == n). If so, it returns 1, indicating a valid assignment.

Otherwise, the function iterates over all topics. For each topic, it checks if:

The student likes the topic (pref [student][topic] == 1).

The topic is not already assigned to another student (! is Assigned (topic)).

If both conditions are true, the function assigns the topic to the student (assigned [student] = topic) and recursively calls itself with the next student (assign Topics (student + 1)).

After the recursive call, the function resets the assignment (assigned [student] = -1) to backtrack and try other topics.

The function returns the total number of valid assignments found by summing the results of the recursive calls.

The main function prints the result of the assign Topics function.

Example Walkthrough

Suppose we have 3 students with the following preferences:

Student	Topic 0	Topic 1	Topic 2
0	1	0	1
1	0	1	1
2	1	1	0

The program will try to assign topics to students as follows:

Student 0: tries Topic 0 (likes it), assigns it, and recursively calls assign Topics with Student 1.

Student 1: tries Topic 1 (likes it), assigns it, and recursively calls assign Topics with Student 2.

Student 2: tries Topic 2 (likes it), assigns it, and returns 1 (valid assignment).

The recursion unwinds, and the program backtracks to try other topics.

The final result is 2, indicating that there are 2 valid assignments of topics to students.