

British Informatics Olympiad Final

30 March – 2 April, 2000

Sponsored by Data Connection & SkillIT

University Places

There are n students applying for places at university and, coincidentally, n places available. Each student has an order of preference for the n places. Interviews having already occurred, each university place has also put the n students in order of preference. Your task is to find a suitable way of pairing the students with the university places.

If, after assigning pairs, there exists a student and university place, both of which prefer each other to their assigned pairing, there is the danger that the university will try to ‘poach’ this student, who will accept (since both will be better of). This is irrespective of the preferences of the other student and place, who may end up better or worse off. An assignment of students to university places is *only* suitable if this situation does not exist.

For example, suppose there are two students (S_1 and S_2) and two university places (P_1 and P_2), where both students prefer P_1 over P_2 , P_1 prefers S_1 over S_2 , and P_2 prefers S_2 over S_1 . The assignment $S_1 - P_2$ & $S_2 - P_1$ is not suitable, since S_1 would prefer to be at P_1 and vice versa. The assignment $S_1 - P_1$ & $S_2 - P_2$ is suitable.

The first line of input will be a single number n ($1 \leq n \leq 100$) indicating the number of students/places. The next n lines will give the preferences for the students, the i^{th} of which containing the preferences for student i . The final n lines will give the preferences for the university places, the i^{th} of which containing the preferences for place i . Both students and places are numbered from 1 to n . A preference list is a permutation of the numbers 1 to n (i. e. each number from 1 to n will occur once and only once in each list), giving the preferred choice at the left and the last choice at the right.

You should output n lines, giving a suitable assignment of places to students. The i^{th} of these lines should be a single integer, indicating the place you assign to student i .

There always exists a matching and you are only required to produce one.

Sample Input

```
5
1 2 3 4 5
2 3 4 5 1
2 1 5 4 3
1 4 5 2 3
1 4 2 3 5
1 2 3 4 5
3 2 1 4 5
4 5 1 2 3
1 2 3 4 5
1 4 5 2 3
```

Sample Output

```
1
4
2
5
3
```