Problem C:
Exam Graph

Source file: `graph.{c | cpp | java}`
Input file: `graph.in`

Your University has been "torturing" students for some years now, scheduling multiple final exams on the same day, as a proper scheduling algorithm has not been designed yet.

The registrar has been offering the raw data of student registrations in the form of (studentID, courseID) pairs, in the hope that one of the wiz student programmers will pick up the challenge and find a solution.

The problem is that these data cannot be used in this form. One has to first create the matrix that corresponds to the (non-weighted) graph where each node represents a course, and each edge connects courses taken together by at least one student.

Input

The input is made of several test cases. Each test case starts with a number *P* which is the number of (studentID, courseID) pairs. *P* lines follow, with the studentID and courseID separated by a single space. The input ends with a test case with *P*=0. The studentID and courseID identifiers are both strings, each having a maximum of 10 characters. The strings are case sensitive, so a course identified as CMP100 is different than cmp100.

Output

For each test case output the adjacency matrix describing the graph. The weights should be separated by a single space, with no trailing spaces at the end of the line. Test case outputs should be separated by a single empty line.
In the construction of the matrix, you should order the courses in ascending alphanumerical order.

Sample Input

```
5
10001 CMP100
10001 CMP101
10002 CMP100
10002 CMP101
10003 CMP102
0
```

Output for Sample Input

```
0 1 0
1 0 0
0 0 0
```