A computing center has ten different computers (numbered 0 to 9) on which applications can run. The computers are not multi-tasking, so each machine can run only one application at any time. There are 26 applications, named A to Z. Whether an application can run on a particular computer can be found in a job description (see below).

Every morning, the users bring in their applications for that day. It is possible that two users bring in the same application; in that case two different, independent computers will be allocated for that application.

A clerk collects the applications, and for each different application he makes a list of computers on which the application could run. Then, he assigns each application to a computer. Remember: the computers are *not* multi-tasking, so each computer must handle at most one application in total. (An application takes a day to complete, so that sequencing i.e. one application after another on the same machine is not possible.)

A job description consists of

1. one upper case letter `A...Z`, indicating the application.
2. one digit `1...9`, indicating the number of users who brought in the application.
3. a blank (space character.)
4. one or more different digits `0...9`, indicating the computers on which the application can run.
5. a terminating semicolon '`;`'.
6. an end-of-line.

Input

The input for your program is a textfile. For each day it contains one or more job descriptions, separated by a line containing only the end-of-line marker. The input file ends with the standard end-of-file marker. For each day your program determines whether an allocation of applications to computers can be done, and if so, generates a possible allocation.

Output

The output is also a textfile. For each day it consists of one of the following:

- ten characters from the set {'`A`' ... '`Z`', '`_`'}, indicating the applications allocated to computers 0 to 9 respectively if an allocation was possible. An underscore '`_`' means that no application is allocated to the corresponding computer.
- a single character '`!`', if no allocation was possible.

Sample Input

```
A4 01234;
Q1 5;
P4 56789;

A4 01234;
Q1 5;
P5 56789;
```

Sample Output

```
AAAA_QPPPP
!
```