

# Dave, the Telecom System Architect, Goes to Elbonia

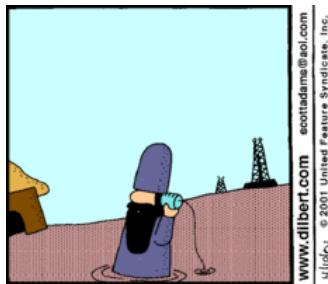
Dave has been tasked with modernizing Elbonia's telecommunication system<sup>1</sup> – no small task given that they currently use tin can phones. He is to replace some of the string links with high-speed rope in order to create a high-speed tin-can phone backbone.

The tin-can backbone must have the following properties:

- All phones must be within one link of the backbone
- The links in the backbone must be connected, i.e. there can be no breaks or gaps in the backbone.

Consider the links in the telecommunication systems shown below. Each labeled node represents a phone, and each edge between the nodes represents a link. The thicker links are links in Dave's proposed backbone. Your job is to determine if the proposed backbone satisfies the requirements of a backbone.

<sup>1</sup>: Elbonia is the fictional country depicted in some Dilbert cartoons.



## Input Format

The input starts with two positive integers  $k$  and  $n$ , both less than 10,000. The number  $k$  represents the number of rope links in the proposed backbone, while the number  $n$  represents the number of string links in the phone system (i.e. non-backbone links). The  $k + n$  lines with the links have the following format:

```
Phone1 Phone2
```

Where

- **Phone1** is an alphanumeric string with the label of the phone at one end of the link
- **Phone2** is an alphanumeric string with the label of the phone at one end of the link

The first  $k$  lines are rope links; the last  $n$  lines are the string links.

## Output Format

The output of the program should a single word, either **valid** if the proposed backbone represents a valid tin-can backbone, or **invalid** if the proposed backbone violates one or more of the requirements of a tin-can backbone.

## Sample Input

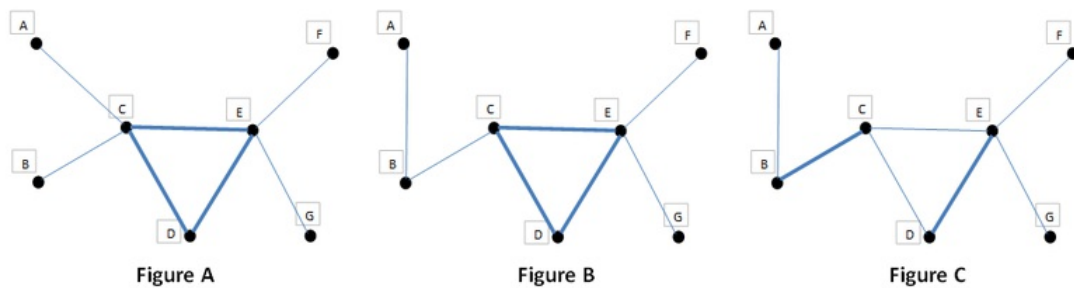
```
3
4
C E
C D
D E
A C
```

B C  
E F  
E G

### Sample Output

valid

### Explanation



The test case above corresponds to the backbone in figure A, which is valid.

If you run your program you will see test cases that correspond to figure B and figure C. Note that the backbone in figure B is not valid because the phone A is not within one link of the backbone. The backbone in figure C is not valid because it is not connected.