4518 Semigroups

A binary operation on a set S is a mapping of every ordered pair of elements from S to another element of S. Addition of integers is such an operation, since the sum of every pair of integers is another integer. But division of integers is not a binary operation, since the result of some division operations is not an integer.

The word *ordered* is important in the definition of a binary operation, since changing the order of the operands may well yield a different result. For example, in integer subtraction, the result of 1 - 2 is certainly not the same as 2 - 1. Binary operations for which the order of the operands is unimportant are called *commutative*. That is, if for some operator '@' we have a @ $b = b$ @ a for all possible values of a and b, then the operator '@' is commutative.

The examples used above, integer addition and subtraction, are defined on infinite sets. But many operations are defined on sets with a finite number of elements. For example, logical operations like and, or, and exclusive or are defined on the set of logical values, false and true. In this problem we are concerned only with operators on finite sets.

A convenient way of documenting an operator is to enumerate the elements of the (finite) set on which it is defined, and to display a table showing the result of the operator. Each row and each column in this table is labeled with a set element, and the result of the operator is given in the body of the table.

For example, consider the logical and operator. The set of elements is, of course {false, true}, and the table might be displayed as follows:

	false	true
false	false	false
true	false	true

A binary operator '@' is said to be associative if, for all possible operands a, b, c we have (a @ b) @ $c = a$ @ (b @ c). As we can see, the logical and operator is associative.

If a binary operation '@' on a set S is associative, then the pair $(S, @)$ forms a semigroup. If the binary operation is also commutative, then the semigroup is also said to be commutative.

In this problem you will be given a description of an operator — that is, the elements of the finite set on which the operation is defined, and the body of the table defining the result of the operator on every pair of operands. Your program must then identify the first of these phrases that applies:

```
is not a binary operation.
is not a semigroup.
is not a commutative semigroup.
is a commutative semigroup.
```

The first statement applies if the result of applying the operator to at least one pair of operands yields a result that is not an element of the finite set on which the operator is defined.

Input

There will be multiple cases to consider. The first line of the input for each case will consist of the elements of the set on which the operator is defined. These will be given as N unique lowercase alphabetic characters; N will be no larger than 10. The next N lines of the input give the body of the table describing the result of applying the binary operator to two operands. Each line will contain N lowercase alphabetic characters. The rows and columns of the table are given in the same order as the elements on the first line of input for the case.

The last case is followed by a line containing only an end of line character. There will be no spaces or tab characters in the input.

Output

For each input case, the output is the case number and one of the phrases specified above. The sample input and output illustrates the appropriate format.

Display a blank line following the output for each case.

Sample Input

```
ft
ff
ft
xy
yx
xz
abc
abc
bca
cba
ab
aa
bb
```

Sample Output

```
Case 1: The operator is a commutative semigroup.

Case 2: The operator is not a binary operation.

Case 3: The operator is not a semigroup.

Case 4: The operator is not a commutative semigroup.
```