Database Dilemma

Tiger IT is a prominent software company in Bangladesh. Mr Kool is currently working as a software engineer there. Mr Kool's boss Mr X has assigned him an interesting project. In that project, there are lots of database stuff Mr Kool will need to do and Mr X will be very angry if he messes up any of them. So, he needs your help to create a simulator for checking the database queries. Can you help him?

In this problem, you will be given some table information. Every table has a unique name and you can assume the name of each column of these tables are unique too. These tables are filled with numeric data. You need to implement JOIN operation on those tables. A JOIN is a means for combining fields from two tables (or more) by using values common to each based upon the join-predicate. The possible join queries are as follows:

Sample 1:

##QUERY FORMAT##

```
SELECT *
FROM <first_table_name>
JOIN <second_table_name>
ON <first_table_name>.<a name of the column from the first table> = <second_table_name>.<a
name of the column of the second table>
```

##QUERY EXAMPLE##

```
SELECT *
FROM table_a
JOIN table_b
ON table_a.column_a = table_b.column_b
```

Sample 2:

##QUERY FORMAT##

```
SELECT *
FROM <first_table_name> <first_table_short_name>
JOIN <second_table_name> <second_table_short_name>
ON <first_table_short_name>.<column_name> = <second_table_short_name>.<column_name>
```

##QUERY EXAMPLE##

```
SELECT *
FROM table_a ta
JOIN table_b tb
ON ta.column_a = tb.column_b
```

Sample 3:

##QUERY FORMAT##

```
SELECT <table_short_name>.<column_name>, <table_short_name>.<column_name>, ...
FROM <first_table_name> <first_table_short_name>
JOIN <second_table_name> <second_table_short_name>
ON <first_table_short_name>.<column_name> = <second_table_short_name>.<column_name>
```

##QUERY EXAMPLE##

```
SELECT ta.column_a1, ta.column_a2, ta.column_a3, tb.column_b1, tb.column_b2
FROM table_a ta
JOIN table_b tb
ON ta.column_a0 = tb.column_b0
```

Input/Output

In the first line of input, you will be given number of test cases ***T***. In every test case, the first line will contain the number of tables ***nT***. Then ***nT*** tables will be described. The first line of each table will contain the ***"table name"***. In the next line you will get two integers ***nC*** and ***nD*** where ***nC*** means number of columns of this table and ***nD*** means number of records this table contains. You can assume that all the table and column names will contain only small english letters, digits and underscore sign (_). You can also assume that the first column of every table will be the primary key of that table. In the next line you will be provided the column names separated by space. Next ***nD*** lines will contain ***nC*** space separated numeric data ***(di)*** of the table according to the column names. After all the table information, you will be provided the number of queries ***nQ*** you need to apply on the tables provided for this test case. Each query will contain four lines of input and you can assume that every query will be one of the sample provided in the problem. Also, there will be a blank link after each query.

You can assume all the table column will contain data and every query is legal. For every test case you need to print ***"Test: d"***, where ***d*** is the test case number. For each query print the result of JOIN operation after printing the column names first. You should print only the columns given in the query and in the appropriate order. If asterisk sign (*) is provided in the query, you need to print all the column names according to the input provided. The rows should be ordered lexicographically. Print blank line after the answer for each query. See the example for more information.

Constraints

1 <= ***T*** <= 10
2<= ***nT*** <= 10
2<= ***nC, nD*** <= 100
0 <= ***di*** <= 10^6
0 <= ***nQ*** <= 50

Sample Input/Output

Sample Input	Output of Sample Input
1 2 table_a 3 3 id_a a1 a2 1 2 3 2 4 5 3 6 7 table_b 3 3 id_b b1 b2 1 2 9 2 10 5 3 12 7 3 SELECT * FROM table_a JOIN table_b ON table_a.a1 = table_b.b1 SELECT * FROM table_a ta JOIN table_b tb ON ta.a2 = tb.b2 SELECT ta.a1, ta.a2, tb.b1 FROM table_a ta JOIN table_b tb ON ta.a2 = tb.b2	Test: 1 id_a a1 a2 id_b b1 b2 1 2 3 1 2 9 id_a a1 a2 id_b b1 b2 2 4 5 2 10 5 3 6 7 3 12 7 a1 a2 b1 4 5 10 6 7 12

Note: Please consider that you need to print the output exactly according to the format provided in the problem. Your code will be judged by using an automated system. So, wrong output format may make your solution wrong.