Google BITWISE 2011 facebook

Dinner with Sachin (Points: 200)

Mr. Bean has arrived to see the India vs. England Cricket World Cup match in Bangalore. The stadium authorities have found a unique way of generating match tickets. They have a starting $m-$digit configuration, $S = [0\ 1\ 2\ ...\ m-1]$ and have designed a set of n rules. Each rule R is a bijective function, $R : \{0, 1, ..., m-1\} \rightarrow \{0, 1, ..., m-1\}$ with the condition that $R(a) = b$ for some $a, b \in \{0, 1, ..., m-1\}$ if and only if $R(b) = a$.

Application of a rule R to a configuration C_0 of size m yields another configuration $C_1 = [R(C_0[0])\ R(C_0[1])\ R(C_0[2])\ ...\ R(C_0[m-1])]$. Applying r rules in some order $R_0 \rightarrow R_1 \rightarrow R_2 \rightarrow ... \rightarrow R_r$ on configuration C_0 means applying R_0 on C_0 to generate configuration C_1, then applying R_1 on C_1 to generate configuration C_2 and so on.

The authorities generate the row verification codes of tickets by applying any number of rules from the set of rules in some order on S. The row in which a spectator has to sit is the minimum number of rules which need to be applied in some order to convert S to his/her row verification code on ticket. Though there are personnel out there to help those who can't find out their rows, the authorities are offering a private dinner with Sachin Tendulkar to the spectator who first finds in which row his seat lies.

Mr. Bean is a big fan of Mr. Tendulkar. He has asked for your help to quickly find his row.

Input Format:

First line contains the number of test cases **T**. For each test case, the first line of input will have two integers **M** and **N** seperated by space. **M** represents the number of digits in row verification code of a ticket and **N** represents the total number of rules in the set. This will be followed by **N** lines of input representing **N** rules. Let $R_0, R_1, R_2, ..., R_{\mathbf{N}-1}$ be the **N** rules. The i^{th} of these **N** lines is a **M** length sequence of distinct numbers $R_i(0), R_i(1), R_i(2), ...,, R_i(\mathbf{M}-1)$ in the range 0 to $\mathbf{M}-1$ seperated by spaces. Each of these sequences will follow the desired property of R. After these **N** lines, the next line contains the number of row verification codes **K**. This is followed by **K** lines, each line is a **M** length sequence of distinct numbers in the range 0 to $\mathbf{M}-1$, representing a possible row verification code.

Limits: $1 \leq \mathbf{T} \leq 5$, $1 \leq \mathbf{M} \leq 1000$, $1 \leq \mathbf{N} \leq 15$, $1 \leq \mathbf{K} \leq 25$.

Output Format:

For each row verification code of all the test cases, output one integer giving the least number of rules (rules of that particular test case) required to generate the row verification code from S, i.e. row corresponding to the row verification code. If there is no possible way of generating the row verification code, output -1.

Sample Input:

```
2

4 4
1 0 3 2
0 1 3 2
2 3 0 1
0 3 2 1
4
0 2 1 3
1 0 2 3
2 1 3 0
0 3 2 1

3 2
1 0 2
2 1 0
3
0 1 2
0 2 1
1 2 0
```

Sample Output:

```
-1
2
3
1
0
-1
2
```

Instructions

- Your program should not print anything other than what is specified in the output format. A program with extraneous output (even a single space) will be treated as incorrect!
- While submitting your code, please select the language carefully *gcc/g++*. Using the wrong language will lead to compiler error.
- The only input/output functions allowed are `printf, scanf, cin, cout`. Perform all read/write operations through stdin/stdout. The solutions will be checked using command line redirection only.

Problem Setter: *Jawaid Iqbal & Shubham Agarwal*