Back to School '16: Contest Practice

In addition to regular schoolwork, **jlsajfj** frequently visits a website where programmers can practice writing algorithms. This website has an extensive list of classical problems which leaves most aspiring programmers satisfied. **jlsajfj**'s favo**u**rite part of the website is the unconventional contest system.

Every month, the website hosts a contest with P problems. Every problem in this contest has Q possible scores, which are integers from 0 to $Q-1$ inclusive. Obviously, a competitor receives exactly one final score for a single problem. A **scoring distribution** is a sequence of P integers where the i^{th} integer is the user's score on the i^{th} problem.

User Y is **equal or better** than user Z when, for each of the P problems, user Y has an equal or better score than user Z. A user's **improvement index** is the number of equal or better participants. This includes oneself, since nobody should have an improvement index of 0!

jlsajfj convinced a friend, **Shinigami**, to try out an amazing contest. However, the website prevents users from knowing the leaderboard position during the contest, as this may influence the users' performance, and further, the website normally posts the results one week later. This does not stop **jlsajfj** from discovering and accessing the well-hidden database on the website. This database contains the scoring distributions of all N users.

Both **jlsajfj** and **Shinigami** want to be famous for implementing a system where you can check your improvement index, so they split the work (fairly unevenly). **jlsajfj** successfully builds a program which determines whether a specific scoring distribution occurred in the contest. But **Shinigami** is stuck trying to calculate the improvement index of a user. In total, there are X valid queries sent to the system at the same time. Help **Shinigami** with his work!

Input Specification

The first line contains the integers P $(1 \leq P \leq 20)$, Q $(2 \leq Q \leq 20)$ and N $(1 \leq N \leq 50\,000)$. It is guaranteed that $2 \leq Q^P \leq 200\,000$.

The following N lines contain a scoring distribution (described above).

The next line contains X $(1 \leq X \leq 50\,000)$, the number of queries that **Shinigami** must process.

In each of the next X lines, there is a scoring distribution. One of the N users is guaranteed to have this distribution.

Output Specification

For each of the X queries, output the improvement index of the user who had this scoring distribution.

Constraints

Subtask 1 [25%]

$N \leq 500$

Subtask 2 [25%]

$P \leq 2$

Subtask 3 [50%]

No further constraints.

Sample Input

```
4 11 5
10 10 10 10
10 10 2 5
10 9 5 0
0 10 0 0
0 0 0 0
6
10 10 10 10
10 10 2 5
10 9 5 0
0 10 0 0
0 0 0 0
10 9 5 0
```

Sample Output

```
1
2
2
3
5
2
```