Car.*

As mentioned before, there are **N** workers in WG’s factory. They are manufacturing cars on a conveyor belt, in a pipeline fashion. Workers are denoted by numbers 1 – leftmost, to **N** - rightmost. Each of the workers does his specific job and requires certain amount of time to complete it.
Production of a single car starts with worker #1 (WG). After he had finished with his part of the job, worker #2 takes over, after him #3... When worker #**N** finishes with his part, the car is finished. WG and his workers have to produce **M** cars and they **must** produce them in order 1 to **M**.
For every worker **i** we know **Ti** - time required for him to do his part of the job. For every car **j** we know factor of assembly complexity **Fj**. Time in minutes for worker **i** to finish his part of he job on the car **j** is computed as a product **TiFj**.
After some worker has finished working on a car, he has to give it to the next worker **instantly**, without any delay (weird company policy). For that reason, the worker receiving the car has to be free (he must not be working on some other car). In order to fulfill this condition, WG has to choose a good timing to start building a new car. To be efficient, he’ll wait **minimum** number of minutes until he is certain that all of the conditions described are met.
Write a program which will, given worker times and factors of complexity for each car, compute total time required for producing all of the cars.

INPUT

First line of input contains space-separated positive integers **N** (1 ≤ **N** ≤ 100 000), number of workers, and **M** (1 ≤ **M** ≤ 100 000), number of cars.
i-th of the following **N** lines contains worker time **Ti** for the worker **i**.
j-th of the following **M** lines contains factor of complexity **Fj** for the car **j**.
These conditions hold: 1 ≤ **Ti** ≤ 10 000, 1 ≤ **Fj** ≤ 10 000.

OUTPUT

First and only line of output has to contain required number of minutes.

Input	Output	Input	Output	Input	Output
3 3 2 1 1 2 1	11	3 3 2 3 3 2 1	29	4 5 3 2 2 2 3 1 2	55

Toy.*

JOE found **N** boxes with various forgotten toys at his attic. There are **M** different toys, numbered 1 through **M**, but each of those can appear multiple times across various boxes.
JOE decided that he will **choose some boxes** in a way that there is **at least one toy of each kind** present, and throw the rest of the boxes away.
Determine the number of ways in which JOE can do this.

INPUT

The first line of input contains two integers **N** and **M** ($1 \le N \le 1\ 000\ 000$, $1 \le M \le 20$).
Each of the following **N** lines contains an integer **Ki** ($0 \le Ki \le M$) followed by **Ki** distinct integers from interval [1, **M**], representing the toys in that box.

OUTPUT

The first and only line of output should contain the requested number of ways modulo 1000000007.

Input	Output	Input	Output	Input	Output
3 3 3 1 2 3 3 1 2 3 3 1 2 3	7	3 3 1 1 1 2 1 3	1	4 5 2 2 3 2 1 2 4 1 2 3 5 4 1 2 4 5	6