

Charlie and the Chocolate Factory

Charlie works in a magical chocolate factory. Packets of its marzipan are made on conveyor belts. To make marzipan perfect, M conveyor belts are used, and the process is as follows.

Each of the M belts has N cells. Charlie first makes a few initial packets and puts them on the cells of the first belt (there may be zero or more packets in a particular cell).

Then the first belt generates a second belt such that each cell of the second belt, at the same time, counts the packets in some five cells of the first belt and creates that number of packets in itself.

For example, the first cell of the second belt sums the packets in the cells 1, 2, 3, 5, 9 of the first belt, the second cell of the second belt sums the packets in the cells 2, 3, 4, 5, 6 of the first belt and so on.

Then, from the second belt, a third belt is generated in the same manner, then the fourth from the third and so on until the M^{th} . Since the number of packets on the belts usually increases, the number of packets in each cell is observed only modulo 10007.

You, as the winner of the golden ticket to the factory, are able to see how the M^{th} belt looks like - that is, how many packets of marzipan there is in each cell. Charlie has also explained to you the production process and now you are wondering how first belt looked like.

Input

The first line of input contains positive integers N ($N \leq 100$) and M ($M < 2^{31}$).

The next N lines describe the process of generating each new belt. In the A^{th} of these lines there are five distinct integers from the interval $[1, N]$, denoting the cells of a previous belt from which the packets are added to the A^{th} cell of a new belt.

The next line contains N integers describing the state of the M^{th} belt (modulo 10007).

Output

Print N integers describing the state of the first belt (modulo 10007). A **unique** solution will exist in all of the test data.

Example

Input:

```
6 3
1 2 3 4 5
1 2 3 4 6
1 2 3 5 6
1 2 4 5 6
1 3 4 5 6
2 3 4 5 6
13 12 12 12 14 12
```

Output:

1 0 0 0 2 0

Explanation of the sample case:

The process goes like this: (1 0 0 0 2 0) - (3 1 3 3 3 2) - (13 12 12 12 14 12)