

Problem C. Collections In Containers

Input file: collections.in
Output file: collections.out
Time limit: 2 seconds
Memory limit: 256 mebibytes

A d -dimensional collection is a set of vectors such that:

- Each coordinate of each vector in the set is a non-negative integer number.
- Let a vector $v = (v_1, \dots, v_d)$ belong to the set. Then every vector $u = (u_1, \dots, u_d)$ such that all u_i are non-negative integers and u does not exceed v coordinate-wise (that is, for every index i from 1 to d the inequality $0 \leq u_i \leq v_i$ must hold) must belong to the set as well.

For example, $\{(0,0), (0,1), (1,0)\}$ and $\{(0,0), (1,0), (2,0), (3,0)\}$ are 2-dimensional collections, while $\{(0,1), (1,0), (1,1)\}$, $\{(0,0), (-1,0)\}$, and $\{(0,0), (0, \frac{1}{2}), (0,1)\}$ are not.

You have to store two identical d -dimensional collections, each of size n . In order to do that, you gathered n identical d -dimensional containers each of which has a capacity given by the *capacity vector* $c = (c_1, \dots, c_d)$. You decided to choose a container for each vector from each collection in such a way that every container would contain exactly one vector from the first collection and exactly one vector from the second collection. However, if two vectors $v = (v_1, \dots, v_d)$ and $u = (u_1, \dots, u_d)$ are placed in the same container, it must be verified that $v_i + u_i \leq c_i$ for every i ; that is, the vector $v + u$ must not exceed the capacity vector c coordinate-wise.

It is also guaranteed that each vector from each collection fits in any container by itself; that is, each vector v of each collection does not exceed the capacity vector c coordinate-wise.

Finding an arrangement to place all those vectors in containers turned out to be hard, so you decided to write a program that will solve this problem.

Input

The first line of input contains two space-separated positive integers n and d ($1 \leq n \cdot d \leq 10^5$) — the size and the dimension of each collection.

The second line of input contains d space-separated integers c_1, \dots, c_d ($0 \leq c_i \leq 10^9$).

Next n lines of input contain description of the collections; i -th of these lines contains d space-separated integers v_{i1}, \dots, v_{id} ($0 \leq v_{ij} \leq 10^9$) — coordinates of i -th vector in the collection. It is guaranteed that the described set of vectors is indeed a d -dimensional collection according to the definition above, and $v_{ij} \leq c_j$.

Output

Output n lines, with two space-separated integers in each — the numbers of vectors from the first and the second collection respectively that must be put in the i -th container. The numeration of vectors in the input is 1-based. Every vector from every collection must be assigned to exactly one container.

If there is more than one possible arrangement, output any one of them. It is guaranteed that there is at least one arrangement that satisfies all the requirements.

Examples

collections.in	collections.out
4 2 1 1 0 0 0 1 1 0 1 1	1 4 2 3 3 2 4 1
4 2 2 1 0 0 1 0 2 0 0 1	1 4 2 2 3 1 4 3