Yet Another Assignment Problem

New semester is coming. As the class monitor, Cathy Yin is going to make necessary preparations. She has *m* jobs to do, *n* classmates are going to help her. Each job requires some classmates working on it for certain time, say the *i*-th classmate must work on the *j*-th job for A_{ij} minutes. As an Oler of great responsibility she wishes to finish all jobs as soon as possible. But a classmate can do only one job at a time, and two classmates can **not** do the same job at the same moment. For example, to decorate the classroom, Alpha must work on it for 3 minutes **plus** Beta works on it for 4 minutes, then one possible assignment will be ABABBAB, taking 7 minutes in total.

Now she is going to make a detailed schedule specifying who is doing what at each moment. Jobs are independent and may be done in any order. Also for each job anyone can do it for arbitrarily long, but **not** longer than the required time A_{ij}. Anyone can be free at any time. Time for certain classmate doing certain work need **not** be consecutive.

As her friend, you are to help her to work out the schedule minimizing the total time needed.

Input

First line of the input contains two positive integers *m*, *n* (1 <= *m*, *n* <= 2000), number of jobs and classmates.

m lines follow, each descibing a job. *i*-th line contains *n* non-nagative integers (<= 10^6), where the *j*-th number is A_{ij}, meaning that the *j*-th classmate has to work on the *i*-th job for A_{ij} minutes as descibed above.

Output

First line contains single integer T, minimum time needed. Next line contains *n* non-negative intergers (<= *m*), giving one possible schedule for the **first** minute, where the *i*-th number specifying the job for the *i*-th classmate to do, and 0 denotes that the corresponding classmate is free.

If there are multiple solutions, any one is accepted.

Example

Input:
2 2
2 5
5 1

Output:
7
1 0

Explanation:

Two jobs are assigned to two classmates, say Lambda and Mu. To clean up the classroom Lambda needs to work for 2 minutes and Mu 5 minutes; and to move desks for new comers Lambda 5 minutes and Mu 1 minute.

One optimal schedule is:

T	Lambda	Mu
0	Tidy	Free
1	Move	Tidy
2	T	M
3	M	T
4	M	T
5	M	T
6	M	T

7 minutes in total. It is obvious that it is impossible to finish it in less than 7 minutes.