A. Flag

time limit per test: 2 seconds
memory limit per test: 64 megabytes
input: standard input
output: standard output

According to a new ISO standard, a flag of every country should have a chequered field $n \times m$, each square should be of one of 10 colours, and the flag should be «striped»: each horizontal row of the flag should contain squares of the same colour, and the colours of adjacent horizontal rows should be different. Berland's government asked you to find out whether their flag meets the new ISO standard.

Input

The first line of the input contains numbers n and m ($1 \le n, m \le 100$), n — the amount of rows, m — the amount of columns on the flag of Berland. Then there follows the description of the flag: each of the following n lines contain m characters. Each character is a digit between `0` and `9`, and stands for the colour of the corresponding square.

Output

Output YES, if the flag meets the new ISO standard, and NO otherwise.

Examples

input

```
3 3
000
111
222
```

output

```
YES
```

input

```
3 3
000
000
111
```

output

```
NO
```

input

```
3 3
000
111
002
```

output

```
NO
```

B. Burglar and Matches

time limit per test: 0.5 second
memory limit per test: 64 megabytes
input: standard input
output: standard output

A burglar got into a matches warehouse and wants to steal as many matches as possible. In the warehouse there are m containers, in the i-th container there are a_i matchboxes, and each matchbox contains b_i matches. All the matchboxes are of the same size. The burglar's rucksack can hold n matchboxes exactly. Your task is to find out the maximum amount of matches that a burglar can carry away. He has no time to rearrange matches in the matchboxes, that's why he just chooses not more than n matchboxes so that the total amount of matches in them is maximal.

Input

The first line of the input contains integer n ($1 \leq n \leq 2 \cdot 10^8$) and integer m ($1 \leq m \leq 20$). The $i + 1$-th line contains a pair of numbers a_i and b_i ($1 \leq a_i \leq 10^8, 1 \leq b_i \leq 10$). All the input numbers are integer.

Output

Output the only number — answer to the problem.

Examples

input

```
7 3
5 10
2 5
3 6
```

output

```
62
```

input

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

output

```
7
```

C. Monitor

time limit per test: 0.5 second
memory limit per test: 64 megabytes
input: standard input
output: standard output

Reca company makes monitors, the most popular of their models is AB999 with the screen size $a \times b$ centimeters. Because of some production peculiarities a screen parameters are integer numbers. Recently the screen sides ratio $x: y$ became popular with users. That's why the company wants to reduce monitor AB999 size so that its screen sides ratio becomes $x: y$, at the same time they want its total area to be maximal of all possible variants. Your task is to find the screen parameters of the reduced size model, or find out that such a reduction can't be performed.

Input

The first line of the input contains 4 integers — a, b, x and y ($1 \leq a, b, x, y \leq 2 \cdot 10^9$).

Output

If the answer exists, output 2 positive integers — screen parameters of the reduced size model. Output `0 0` otherwise.

Examples

input

```
800 600 4 3
```

output

```
800 600
```

input

```
1920 1200 16 9
```

output

```
1920 1080
```

input

```
1 1 1 2
```

output

```
0 0
```

D. Logging

time limit per test: 1 second
memory limit per test: 64 megabytes
input: standard input
output: standard output

The main server of Gomble company received a log of one top-secret process, the name of which can't be revealed. The log was written in the following format: «`[date:time]: message`», where for each «`[date:time]`» value existed not more than 10 lines. All the files were encoded in a very complicated manner, and only one programmer — Alex — managed to decode them. The code was so complicated that Alex needed four weeks to decode it. Right after the decoding process was finished, all the files were deleted. But after the files deletion, Alex noticed that he saved the recordings in format «`[time]: message`». So, information about the dates was lost. However, as the lines were added into the log in chronological order, it's not difficult to say if the recordings could appear during one day or not. It is possible also to find the minimum amount of days during which the log was written.

So, to make up for his mistake Alex has to find the minimum amount of days covered by the log. Note that Alex doesn't have to find the minimum amount of days between the beginning and the end of the logging, he has to find the minimum amount of dates in which records could be done. (See Sample test 2 for further clarifications).

We should remind you that the process made not more than 10 recordings in a minute. Consider that a midnight belongs to coming day.

Input

The first input line contains number n ($1 \le n \le 100$). The following n lines contain recordings in format «`[time]: message`», where time is given in format «`hh:mm x.m.`». For `hh` two-digit numbers from `01` to `12` are used, for `mm` two-digit numbers from `00` to `59` are used, and `x` is either character «`a`» or character «`p`». A message is a non-empty sequence of Latin letters and/or spaces, it doesn't start or end with a space. The length of each message doesn't exceed 20.

Output

Output one number — the minimum amount of days covered by the log.

Examples

input

```
5
[05:00 a.m.]: Server is started
[05:00 a.m.]: Rescan initialized
[01:13 p.m.]: Request processed
[01:10 p.m.]: Request processed
[11:40 p.m.]: Rescan completed
```

output

```
2
```

input

```
3
[09:00 a.m.]: User logged in
[08:00 a.m.]: User logged in
[07:00 a.m.]: User logged in
```

output

```
3
```

Note

Formally the 12-hour time format is described at:

- `http://en.wikipedia.org/wiki/12-hour_clock`.

The problem authors recommend you to look through these descriptions before you start with the problem.

E. Fish

time limit per test: 3 seconds
memory limit per test: 128 megabytes
input: standard input
output: standard output

n fish, numbered from 1 to n, live in a lake. Every day right one pair of fish meet, and the probability of each other pair meeting is the same. If two fish with indexes i and j meet, the first will eat up the second with the probability a_{ij}, and the second will eat up the first with the probability $a_{ji} = 1 - a_{ij}$. The described process goes on until there are at least two fish in the lake. For each fish find out the probability that it will survive to be the last in the lake.

Input

The first line contains integer n ($1 \le n \le 18$) — the amount of fish in the lake. Then there follow n lines with n real numbers each — matrix a. a_{ij} ($0 \le a_{ij} \le 1$) — the probability that fish with index i eats up fish with index j. It's guaranteed that the main diagonal contains zeros only, and for other elements the following is true: $a_{ij} = 1 - a_{ji}$. All real numbers are given with not more than 6 characters after the decimal point.

Output

Output n space-separated real numbers accurate to not less than 6 decimal places. Number with index i should be equal to the probability that fish with index i will survive to be the last in the lake.

Examples

input

```
2
0 0.5
0.5 0
```

output

```
0.500000 0.500000
```

input

```
5
0 1 1 1 1
0 0 0.5 0.5 0.5
0 0.5 0 0.5 0.5
0 0.5 0.5 0 0.5
0 0.5 0.5 0.5 0
```

output

```
1.000000 0.000000 0.000000 0.000000 0.000000
```