A. Football

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

One day Vasya decided to have a look at the results of Berland 1910 Football Championship’s finals. Unfortunately he didn't find the overall score of the match; however, he got hold of a profound description of the match's process. On the whole there are n lines in that description each of which described one goal. Every goal was marked with the name of the team that had scored it. Help Vasya, learn the name of the team that won the finals. It is guaranteed that the match did not end in a tie.

Input

The first line contains an integer n ($1 \le n \le 100$) — the number of lines in the description. Then follow n lines — for each goal the names of the teams that scored it. The names are non-empty lines consisting of uppercase Latin letters whose lengths do not exceed 10 symbols. It is guaranteed that the match did not end in a tie and the description contains no more than two different teams.

Output

Print the name of the winning team. We remind you that in football the team that scores more goals is considered the winner.

Examples

input

```
1
ABC
```

output

```
ABC
```

input

```
5
A
ABA
ABA
A
A
```

output

```
A
```

B. Letter

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

Vasya decided to write an anonymous letter cutting the letters out of a newspaper heading. He knows heading s_1 and text s_2 that he wants to send. Vasya can use every single heading letter no more than once. Vasya doesn't have to cut the spaces out of the heading — he just leaves some blank space to mark them. Help him; find out if he will manage to compose the needed text.

Input

The first line contains a newspaper heading s_1. The second line contains the letter text s_2. s_1 и s_2 are non-empty lines consisting of spaces, uppercase and lowercase Latin letters, whose lengths do not exceed 200 symbols. The uppercase and lowercase letters should be differentiated. Vasya does not cut spaces out of the heading.

Output

If Vasya can write the given anonymous letter, print `YES`, otherwise print `NO`

Examples

input

```
Instead of dogging Your footsteps it disappears but you dont notice anything
where is your dog
```

output

```
NO
```

input

```
Instead of dogging Your footsteps it disappears but you dont notice anything
Your dog is upstears
```

output

```
YES
```

input

```
Instead of dogging your footsteps it disappears but you dont notice anything
Your dog is upstears
```

output

```
NO
```

input

```
abcdefg hijk
k j i h g f e d c b a
```

output

```
YES
```

C. Lucky Tickets

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Vasya thinks that lucky tickets are the tickets whose numbers are divisible by 3. He gathered quite a large collection of such tickets but one day his younger brother Leonid was having a sulk and decided to destroy the collection. First he tore every ticket exactly in two, but he didn’t think it was enough and Leonid also threw part of the pieces away. Having seen this, Vasya got terrified but still tried to restore the collection. He chose several piece pairs and glued each pair together so that each pair formed a lucky ticket. The rest of the pieces Vasya threw away reluctantly. Thus, after the gluing of the $2t$ pieces he ended up with t tickets, each of which was lucky.

When Leonid tore the tickets in two pieces, one piece contained the first several letters of his number and the second piece contained the rest.

Vasya can glue every pair of pieces in any way he likes, but it is important that he gets a lucky ticket in the end. For example, pieces 123 and 99 can be glued in two ways: 12399 and 99123.

What maximum number of tickets could Vasya get after that?

Input

The first line contains integer n ($1 \le n \le 10^4$) — the number of pieces. The second line contains n space-separated numbers a_i ($1 \le a_i \le 10^8$) — the numbers on the pieces. Vasya can only glue the pieces in pairs. Even if the number of a piece is already lucky, Vasya should glue the piece with some other one for it to count as lucky. Vasya does not have to use all the pieces. The numbers on the pieces an on the resulting tickets may coincide.

Output

Print the single number — the maximum number of lucky tickets that will be able to be restored. Don't forget that every lucky ticket is made of exactly two pieces glued together.

Examples

input

```
3
123 123 99
```

output

```
1
```

input

```
6
1 1 1 23 10 3
```

output

```
1
```

D. Journey

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

The territory of Berland is represented by a rectangular field $n \times m$ in size. The king of Berland lives in the capital, located on the upper left square $(1, 1)$. The lower right square has coordinates (n, m). One day the king decided to travel through the whole country and return back to the capital, having visited every square (except the capital) exactly one time. The king must visit the capital exactly two times, at the very beginning and at the very end of his journey. The king can only move to the side-neighboring squares. However, the royal advise said that the King possibly will not be able to do it. But there is a way out — one can build the system of one way teleporters between some squares so that the king could fulfill his plan. No more than one teleporter can be installed on one square, every teleporter can be used any number of times, however every time it is used, it transports to the same given for any single teleporter square. When the king reaches a square with an installed teleporter he chooses himself whether he is or is not going to use the teleport. What minimum number of teleporters should be installed for the king to complete the journey? You should also compose the journey path route for the king.

Input

The first line contains two space-separated integers n and m ($1 \le n, m \le 100, 2 \le\ n \cdot m$) — the field size. The upper left square has coordinates $(1, 1)$, and the lower right square has coordinates of (n, m).

Output

On the first line output integer k — the minimum number of teleporters. Then output k lines each containing 4 integers $x_1\ y_1\ x_2\ y_2$ ($1 \le x_1, x_2 \le n, 1 \le y_1, y_2 \le m$) — the coordinates of the square where the teleporter is installed (x_1, y_1), and the coordinates of the square where the teleporter leads (x_2, y_2).

Then print $nm + 1$ lines containing 2 numbers each — the coordinates of the squares in the order in which they are visited by the king. The travel path must start and end at $(1, 1)$. The king can move to side-neighboring squares and to the squares where a teleporter leads. Besides, he also should visit the capital exactly two times and he should visit other squares exactly one time.

Examples

input

```
2 2
```

output

```
0
1 1
1 2
2 2
2 1
1 1
```

input

```
3 3
```

output

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

E. Race

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Today s kilometer long auto race takes place in Berland. The track is represented by a straight line as long as s kilometers. There are n cars taking part in the race, all of them start simultaneously at the very beginning of the track. For every car is known its behavior — the system of segments on each of which the speed of the car is constant. The j-th segment of the i-th car is pair $(v_{i,j}, t_{i,j})$, where $v_{i,j}$ is the car's speed on the whole segment in kilometers per hour and $t_{i,j}$ is for how many hours the car had been driving at that speed. The segments are given in the order in which they are "being driven on" by the cars.

Your task is to find out how many times during the race some car managed to have a lead over another car. A lead is considered a situation when one car appears in front of another car. It is known, that all the leads happen instantly, i. e. there are no such time segment of positive length, during which some two cars drive "together". At one moment of time on one and the same point several leads may appear. In this case all of them should be taken individually. Meetings of cars at the start and finish are not considered to be counted as leads.

Input

The first line contains two integers n and s ($2 \le n \le 100, 1 \le s \le 10^6$) — the number of cars and the length of the track in kilometers. Then follow n lines — the description of the system of segments for each car. Every description starts with integer k ($1 \le k \le 100$) — the number of segments in the system. Then k space-separated pairs of integers are written. Each pair is the speed and time of the segment. These integers are positive and don't exceed 1000. It is guaranteed, that the sum of lengths of all segments (in kilometers) for each car equals to s; and all the leads happen instantly.

Output

Print the single number — the number of times some car managed to take the lead over another car during the race.

Examples

input

```
2 33
2 5 1 2 14
1 3 11
```

output

```
1
```

input

```
2 33
2 1 3 10 3
1 11 3
```

output

```
0
```

input

```
5 33
2 1 3 3 10
1 11 3
2 5 3 3 6
2 3 1 10 3
2 6 3 3 5
```

output

```
2
```