TASK	BUKA	BAZEN	NERED	CUSKIJA	DOSTAVA	SLICICE
input	standard input					
output	standard output					
time limit	1 second	1 second	1 second	1 second	3 seconds	3 seconds
memory limit	32 MB	32 MB	32 MB	32 MB	128 MB	32 MB
points	**30**	**60**	**60**	**100**	**120**	**130**
	500					

Quite often there is substantial noise in the classroom during class. Instead of paying attention to what the teacher is saying, the students rather discuss the economic crisis or Croatia's joining the European Union.

The biggest noise often occurs when the students are idle, so teachers will give them tasks with a lot of work to calm them down. In one type of such task, the student must calculate the result of an arithmetic operation on two large numbers.

The arithmetic operations we will consider are adding and multiplication. The operands will be **powers of 10** with no more than 100 digits.

Write a program that calculates the result of the operation.

INPUT

The first line contains a positive integer A, the first operand.

The second line either the character '+' or '*', representing addition or multiplication.

The third line contains a positive integer B, the second operant.

The integers A and B will be powers of 10 and consist of at most 100 digits.

OUTPUT

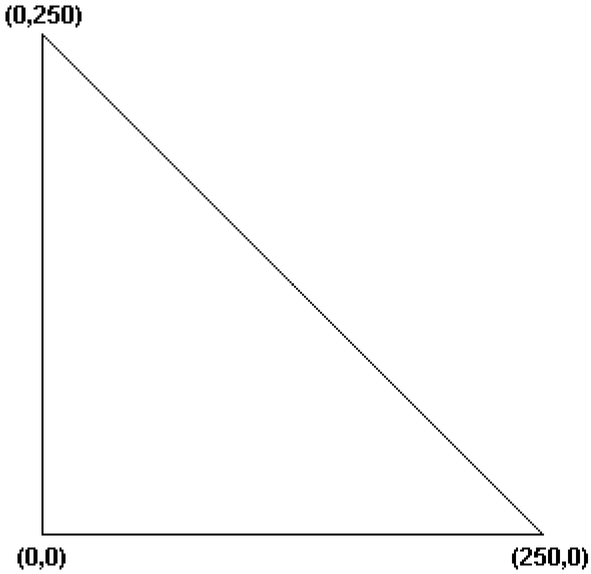
Output the result of the operation.

EXAMPLES

input 1000 * 100 **output** 100000	**input** 10000 + 10 **output** 10010	**input** 10 + 1000 **output** 1010	**input** 1 * 1000 **output** 1000

Mirko and Slavko have built a grand new pool in front of their weekend getaway.

The pool is an isosceles right triangle, with legs of length 250.



Everything was perfect, girls were coming, the DJ was coming, and the parties were wild.

But, a problem arose when they started serving food at the parties. Mirko is vegetarian, while Slavko thinks a party without sausages is no party at all. Therefore they had to divide the pool into two parts.

The pool is placed into a coordinate system (as in the figure above) and divided into two parts of **equal area** by a line segment with **both endpoints on the edges** of the pool.

Write a program that, given one endpoint of the dividing line segment, calculates the coordinates of the other endpoint.

INPUT

The first line contains two integers, the coordinates of one endpoint of the dividing line segment.

That point will be on one of the edges of the pool.

OUTPUT

Output the coordinates of the other endpoint, rounded to two decimal digits.

EXAMPLES

input 0 0 **output** 125.00 125.00	**input** 230 20 **output** 0.00 114.13	**input** 0 40 **output** 148.81 101.19

In the nearby kindergarten they recently made up an attractive game of strength and agility that kids love.

The surface for the game is a large flat area divided into N×N squares.

The children lay large spongy cues onto the surface. The sides of the cubes are the same length as the sides of the squares. When a cube is put on the surface, its sides are aligned with some square. A cube may be put **on another cube** too.

Kids enjoy building forts and hiding them, but they always leave behind a huge mess. Because of this, prior to closing the kindergarten, the teachers rearrange **all** the cubes so that they occupy a rectangle on the surface, with **exactly one** cube on every square in the rectangle.

In one moving, a cube is taken off the top of a square to the top of any other square.

Write a program that, given the state of the surface, calculates the smallest number of moves needed to arrange all cubes into a rectangle.

INPUT

The first line contains the integers N and M ($1 \le N \le 100$, $1 \le M \le N^2$), the dimensions of the surface and the number of cubes currently on the surface.

Each of the following M lines contains two integers R and C ($1 \le R, C \le N$), the coordinates of the square that contains the cube.

OUTPUT

Output the smallest number of moves. A solution will always exist.

EXAMPLES

input 3 2 1 1 1 1 **output** 1	**input** 4 3 2 2 4 4 1 1 **output** 2	**input** 5 8 2 2 3 2 4 2 2 4 3 4 4 4 2 3 2 3 **output** 3

In the first example, it suffices to move one of the cubes from (1, 1) to (1, 2) or (2, 1).
In the third example, a cube is moved from (2, 3) to (3, 3), from (4, 2) to (2, 5) and from (4, 4) to (3, 5).

Rearrange the given array of integers so that the sum of two adjacent elements is never divisible by three.

INPUT

The first line contains an integer N (1 ≤ N ≤ 10000), the number of elements in the array.

The second line contains the elements of the array separated by single spaces. The elements will be positive integers less than 1000000.

OUTPUT

If any valid rearrangement exists, output it on a single line. Otherwise, output "impossible".

EXAMPLES

input	input	input	input
3 1 2 3	5 4 6 3 9 8	6 3 7 6 4 2 8	3 3 12 9
output	**output**	**output**	**output**
2 3 1	3 4 6 8 9	3 7 4 6 2 8	impossible

Little Ivica recently got a job delivering pizzas for the most popular pizzeria in town.

At the start of his work day, he receives a list with the locations to which he needs to deliver pizzas, **in order** in which the locations are given.

The city is divided into R×C cells. The rows are numbered 1 through R, columns 1 through C.

From every cell, it is possible to move to neighbouring cells to the left and right. Moving up or down is only allowed in the first and last columns (columns 1 and C).

The pizzeria is in the top left corner (1, 1) and this is the location Ivica starts from. Ivica takes with him all the pizzas he will deliver that day so he does not have to return to the pizzeria between deliveries or after the last delivery.

For each location in the city, Ivica knows how much time he will spend every time he is in it (trying to get through the intersection, for example).

Write a program that calculates the smallest amount of time for Ivica to deliver all the pizzas.

INPUT

The first line contains the integers R and C ($1 \le R \le 2000$, $1 \le C \le 200$), the dimensions of the city.

Each of the following R lines contains C integers. These are the times Ivica spends every time he enters a location. The times will be integers between 0 and 5000, inclusive.

The next line contains an integer D ($1 \le D \le 200000$), the number of pizza deliveries that day. (No, it's not unrealistically large at all.)

Each of the following D lines contains two integers A and B ($1 \le A \le R$, $1 \le B \le C$), the location to which a pizza must be delivered. The pizzas are given in the order in which they must be delivered. No location will be given twice in a row.

OUTPUT

Output the smallest amount of time for Ivica to deliver all the pizzas.

SCORING

In test cases worth 70% of points, R will be at most 250.

EXAMPLES

input	input
3 3 1 8 2 2 3 2 1 0 1 3 1 3 3 3 2 2	2 5 0 0 0 0 0 1 4 2 3 2 4 1 5 2 2 2 5 2 1
output	**output**
17	9

In the first example, the shortest path goes through the following locations:
(1, 1), (2, 1), (3, 1), (3, 2), (3, 3), (2, 3), **(1, 3)**, (2, 3), **(3, 3)**, (2, 3) and **(2, 2)**.
The locations in bold show where Mirko made deliveries.

The total time for the deliveries is 1+2+1+0+1+2+2+2+1+2+3=17.

After their pool burst, Mirko and Slavko started collecting cards. In their neighbourhood, card collection is taken seriously and there are strict rules for the purchase and trading of cards.

Purchasing cards is **always** done by two children together. Each of them gives half the required funds and **two** cards are bought. Then they race to the fountain downtown, the winner getting both cards. If they arrive at the exact same time, each of them gets one of the cards.

At first the rules performed okay, but soon accusations started that some kids could not have acquired their cards only through such purchases.

One day all the children met to figure out if there were irregularities. They managed to agree on the exact number of cards each of them currently has. They also made a **partial** list of who went to the store with whom, but they do not know who won any of the races and took the cards on those occasions.

Write a program that determines who participated in **all** purchases that were made and who won the subsequent races so that after all the purchases, the cards counts correspond to the given counts. Assume that before any purchases, the children had no cards.

If there are multiple possible solutions, output any of them.

INPUT

The first line contains the integers N and M ($1 \leq N \leq 100$, $0 \leq M \leq 1000$), the number of children and the number of purchases they recall. The children are labeled 1 through N.

The second line contains N integers, the number of cards each child currently has.

The following M lines contain two integers each, the labels of the children who made the purchase.

OUTPUT

On the first line, output the total number of purchases.

Each of the following lines should describe one purchase. The description of a purchase consists of three numbers: the labels of children that made the purchase and the number 0, 1 or 2, how many cards the first child got after the race.

Note: A solution will always exist, although not necessarily unique. The total number of purchases will be at most 1000.

EXAMPLES

<table>
<tr>
<td>

input

```
2 3
5 1
1 2
1 2
1 2
```

output

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

</td>
<td>

input

```
4 3
5 3 1 1
1 3
2 3
4 1
```

output

```
5
1 3 1
2 3 2
4 1 0
2 4 1
1 3 2
```

</td>
<td>

input

```
5 0
3 0 2 4 1
```

output

```
5
1 2 2
1 3 1
4 2 2
3 4 0
3 5 1
```

</td>
</tr>
</table>

In the first example, there are only two children, labeled 1 and 2. The first child should end up with five cards, the second with one.

After the first purchase, each of the children got one card.

After the second and third purchases, the first child got both cards both times.