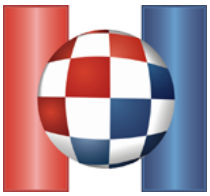


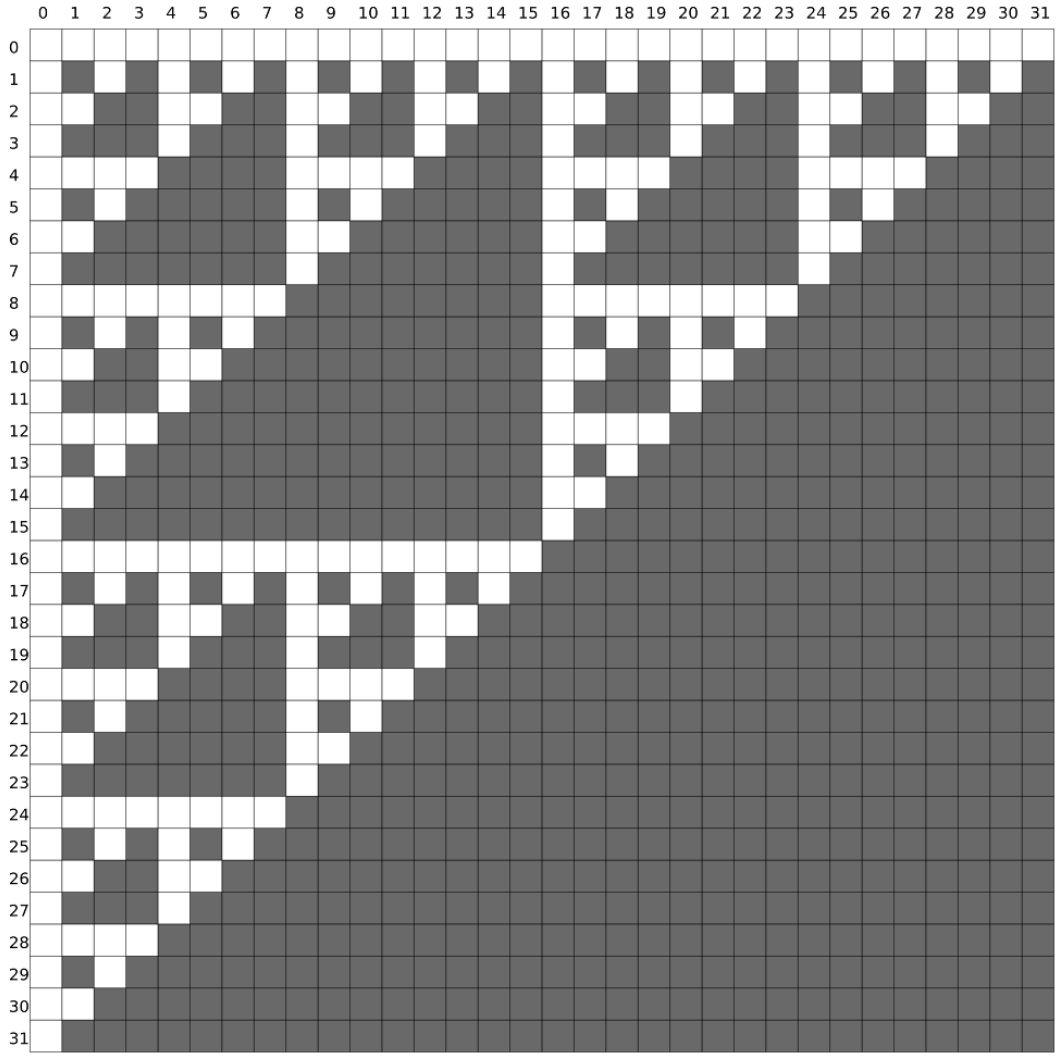


| TASK | ČVENK | KOVANICE | OGLEDALA | SIR |
|---|---|---|---|---|
| **input** | standard input (*stdio*) | | | |
| **output** | standard output (*stdout*) | | | |
| **time limit** | 3 seconds | 2 seconds | 4 seconds | 1 second |
| **memory limit** | 512 MB | 512 MB | 512 MB | 512 MB |
| **points** | **100** | **100** | **100** | **100** |
| | **total 400** | | | |

A group of Czech tourists is walking in a labyrinth of a strange self-similar shape. The ground plan of the labyrinth is a Sierpinski triangle – a fractal structure named after the Polish mathematician Wacław Sierpiński.

The labyrinth consists of a billion rows numbered from 0 to $10^9 - 1$ from top to bottom, and a billion columns numbered from 0 to $10^9 - 1$ from left to right. The fields in the labyrinth can be either free or blocked.

The field in row $X$ and column $Y$ is free if the result of the bitwise '*and*' operation on the numbers $X$ and $Y$ is equal to zero, otherwise it is blocked. In other words, a field is blocked if, when $X$ and $Y$ are switched to binary, there is an integer $k$ such that the $k^{\text{th}}$ digit from the right of the number $X$ and the $k^{\text{th}}$ digit from the right of the number $Y$ are equal to 1.




*The first 32 rows and columns of the labyrinth. The blocked fields are colored in black.*

The Czech tourists are tired from a long day of wandering and would like to meet up in a free field and exchange experiences. In each step, one tourist can jump to one of the adjacent free fields (up, down, left or right).

Write a programme that will, based on the current tourists' locations, determine **minimum total number of steps necessary** in order for **all the tourists to meet in the same field**.

## INPUT

The first line of input contains an integer $N$ – the number of tourists. Each of the following $N$ lines contains two integers $R_i$ and $S_i$ – the row and column of the field where the $i^{\text{th}}$ tourist is located.

All the tourists are located in free fields, and it is possible that there are multiple tourists in the same field.

## OUTPUT

The first and only line of output must contain the required minimum number of steps.

**Please note:** We recommend that you use a 64-bit integer data type (`int64` in Pascal, `long long` in C/C++).

## SCORING

| subtask | subscore | constraints |
|---|---|---|
| 1 | 17 | $N = 2$<br>$0 \leqslant R_K, S_K < 10^9$ |
| 2 | 21 | $2 \leqslant N \leqslant 100$<br>$0 \leqslant R_K, S_K < 10^9$ |
| 3 | 22 | $2 \leqslant N \leqslant 10^5$<br>$0 \leqslant R_K, S_K < 500$ |
| 4 | 40 | $2 \leqslant N \leqslant 10^5$<br>$0 \leqslant R_K, S_K < 10^9$ |

## SAMPLE TESTS

**input**

```
2
2 1
4 3
```

**output**

```
6
```

**input**

```
6
2 5
3 4
8 7
9 6
10 5
11 4
```

**output**

```
50
```

**Clarification of the first example:** One of the fields where the brave Czech tourists could have met is (2, 0).

**Clarification of the second example:** One of the fields where the playful Czech tourists could have met is (8, 4).

Mirko paid a touristic visit to a land far away where banknotes aren't used, but only coins. More precisely, the country has $N$ types of coins in circulation. Their names are, respectively, 'K1', 'K2', 'K3', ..., 'K$N$'. The coins are of the same size and shape, but different weights. 'K1' is the lightest type of coin, 'K2' is the second lightest and so on until the heaviest type 'KN'.

Mirko has $M$ coins in his pocket, but he doesn't know which one is of which type. In order to determine that, he only has a simple balance scale at his disposal.

Initially, Mirko marked his unknown coins with numbers from 1 to $M$ and after that performed $V$ weighings. In each weighing, he put one coin on one side of the scale, and another coin on the other side of the scale. Then he saw whether the two coins weigh equally, and if they don't, which one is heavier.

Write a programme that will, based on the weighing results, determine the type of coin for each coin for which it is possible to determine it uniquely.

## INPUT

The first line of input contains integers $N$, $M$ and $V$ – the number of types of coins in the country, the number of coins in Mirko's pocket and the number of weighings.
Each of the following $V$ lines contains the result of one weighing in the form of $ACB$ where $A$ and $B$ are different integers less than or equal to $M$, and $C$ is the character '$=$' (equal) or '$<$' (less).
There is no space between the numbers and character $C$. The result of one weighing tells us that Mirko's coin marked with $A$ is of equal weight as the coin marked with $B$ or lighter than it.

The weighing results won't be contradictory.

## OUTPUT

Output $M$ lines. The $i^{\text{th}}$ line must contain the type of coin marked with $i$ – a sequence of characters of the form 'K$X$' where $X$ is an integer between 1 and $N$.
If it isn't possible to uniquely determine the weight of the coin marked with $i$, output the character '?' in the $i^{\text{th}}$ line.

## SCORING

| subtask | subscore | constraints |
|---|---|---|
| 1 | 10 | $N = 2$, $1 \leqslant M \leqslant 1\,000$ |
| 2 | 40 | $N = 2$, $1 \leqslant M \leqslant 300\,000$ |
| 3 | 10 | $1 \leqslant M \leqslant 1\,000$ |
| 4 | 40 | $1 \leqslant M \leqslant 300\,000$ |

In all subtasks, it will hold $2 \leqslant N \leqslant 300\,000$ and $1 \leqslant V \leqslant 300\,000$.

## SAMPLE TESTS

**input**

```
3 5 3
1<2
2<4
3=5
```

**output**

```
K1
K2
?
K3
?
```

**input**

```
2 7 6
1=2
2=3
2=7
3<4
4=5
4=6
```

**output**

```
K1
K1
K1
K2
K2
K2
K1
```

There are $M$ modern washbasins numbered from 1 to $M$ from left to right in the ladies' room of an enormous shopping centre.

Currently, there are $N$ ladies in the restroom (numbered from 1 to $N$) and they've occupied the washbasins with numbers $A_1, A_2, \ldots, A_N$. Soon there will be $M - N$ more ladies arriving (numbered from $N + 1$ to $M$ in the order of arrival) that will occupy all the remaining available washbasins. The ladies arriving primarily want their privacy, so each of them always chooses an available washbasin by the following procedure:

- First she finds the largest consecutive series of available washbasins. If there is more than one, she chooses the leftmost.
- After that, she occupies the middle washbasin in the chosen series. If the series is of even length, she chooses the left of the two middle washbasins.

It is safe to assume that none of the ladies will leave the washbasin in the foreseeable future.

Write a programme that will, for each of $Q$ given integers $B_i$, determine the label of the washbasin that will be occupied by the lady marked with $B_i$.

## INPUT

The first line of input contains integers $M$, $N$ and $Q$ – the number of washbasins, the initial number of ladies in the restroom and the number of ladies for which you need to determine which washbasin they will use.

The second line of input contains $N$ space-separated integers, the $i^{\text{th}}$ of those numbers is $A_i$ – the label of the washbasin that is being used by lady $i$.

The third line of input contains $Q$ space-separated integers, the $i^{\text{th}}$ of those numbers is $B_i$ – the label of the lady for which we want to know which washbasin she will use.

The first $N$ ladies in the restroom haven't necessarily chosen their washbasins using the procedure described in the task.

The arrays $A$ and $B$ are strictly increasing. In other words, it holds $1 \leqslant A_1 < A_2 < \ldots < A_N \leqslant M$ and $1 \leqslant B_1 < B_2 < \ldots < B_Q \leqslant M$.

## OUTPUT

Output $Q$ lines. The $i^{\text{th}}$ line must contain the label of the washbasin that will be used by the lady labeled $B_i$.

**Please note:** We recommend that you use a 64-bit integer data type (`int64` in Pascal, `long long` in C/C++).

## SCORING

| subtask | subscore | constraints |
|---|---|---|
| 1 | 19 | $1 \leqslant M \leqslant 300\,000$ |
| 2 | 22 | $1 \leqslant M \leqslant 10^{14}$, $B_Q \leqslant 300\,000$ |
| 3 | 59 | $1 \leqslant M \leqslant 10^{14}$ |

In all subtasks, it will hold $1 \leqslant Q \leqslant 10^5$, $N \leqslant M$ and $1 \leqslant N \leqslant 10^5$.

## SAMPLE TESTS

**input**

```
7 1 4
4
2 3 4 5
```

**output**

```
2
6
1
3
```

**input**

```
10 2 4
2 8
1 3 5 8
```

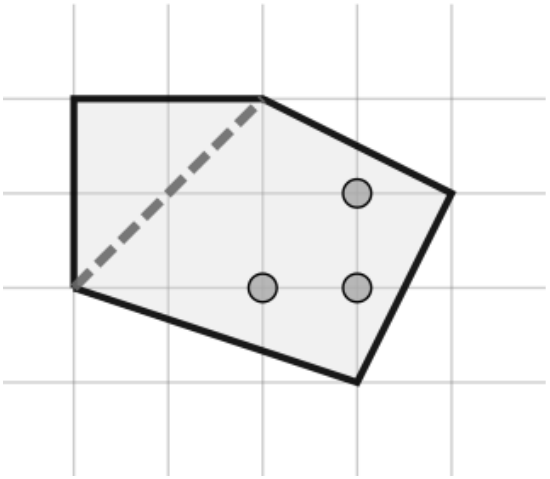
**output**

```
2
5
6
4
```

**Clarification of the second example:** The first two ladies occupy the washbasins 2 and 8 (respectively). The remaining ladies occupy the washbasins 5, 3, 6, 9, 1, 4, 7 and 10, respectively.

Krešo has bought some delicious cheese with peppers, but Stjepan doesn't really like peppers so he's trying to cut a piece that doesn't contain any peppers.

Krešo's cheese has the shape of a convex polygon, and each pepper is one point on the inside of the polygon. Stjepan cuts the cheese **exactly once** in such a way that he chooses two vertices of the polygon that **are not adjacent** and cuts along the line segment connecting them. Stjepan then takes the freshly cut part without any peppers (on the inside nor on the edges).



*The image corresponds to the first sample test. The dotted line marks Stjepan's cut.*

Write a programme that will determine whether Stjepan can cut a piece of cheese without any peppers. If he can, determine **the maximum possible area** of the piece of cheese without peppers that Stjepan can cut.

## INPUT

The first line of input contains the integer $N$ – the number of vertices of the convex polygon representing the cheese.
Each of the following $N$ lines contains two integers $X_i$ and $Y_i$ – coordinates of the $i^{\text{th}}$ polygon vertex.
The following line contains the integer $M$ – the number of peppers.
Each of the following $M$ lines contains two integers $X_i$ and $Y_i$ – coordinates of the $i^{\text{th}}$ peppers.

The polygon vertices are given in counter-clockwise order and form a convex polygon. None two adjacent edges are parallel.

All peppers are located in different coordinates in the inside of the polygon (they will not be located on the edge or outside of the polygon).

## OUTPUT

The first and only line of output must contain an integer – twice the maximum possible area (the double area is always going to be an integer).
If it isn't possible to cut a piece of cheese without peppers in one move, output 0.

## SCORING

| subtask | subscore | constraints |
|---|---|---|
| 1 | 12 | $4 \leqslant N \leqslant 300$, $1 \leqslant M \leqslant 300$ |
| 2 | 39 | $4 \leqslant N \leqslant 3\,000$, $1 \leqslant M \leqslant 3\,000$ |
| 3 | 49 | $4 \leqslant N \leqslant 300\,000$, $1 \leqslant M \leqslant 300\,000$ |

In all subtasks, the coordinates will be from the interval $[-10^9, 10^9]$.

## SAMPLE TESTS

**input**

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

**output**

```
4
```

**input**

```
6
-3 3
-3 -4
-2 -5
2 -5
3 -4
3 3
7
1 0
0 -1
0 -3
2 0
0 0
0 2
-1 0
```

**output**

```
10
```

**input**

```
6
0 3
-1 2
-1 -2
0 -3
1 -2
1 2
1
0 0
```

**output**

```
4
```

**Clarification of the second example:** Stjepan cuts from vertex 2 to vertex 5.
**Clarification of the third example:** Stjepan cuts from vertex 1 to vertex 3.