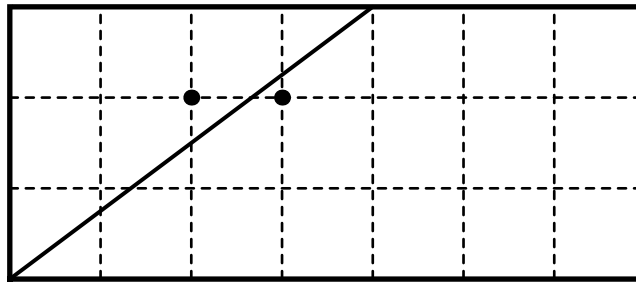


Problem A. Anniversary Cake

Input file: anniversary.in
Output file: anniversary.out
Time limit: 2 seconds
Memory limit: 256 megabytes

Two students, Adam and Anton, are celebrating two-year anniversary of not passing their Math Logic exam. After very careful search in a local supermarket, they bought a rectangular cake with integer dimensions and two candles.

Later in the campus Adam put the candles into different integer points of the cake and gave a knife to Anton to cut the cake. The cut should start and end at integer points at the edges of the cake, and it should not touch the candles. Also each piece should have exactly one candle at it. Please, help Anton to find the starting and ending points of the cut.



A 7×3 cake and two candles at $(2, 2)$ and $(3, 2)$.
Anton can cut this cake through $(0, 0)$ and $(4, 3)$.

Input

The single line of the input contains six integers: w, h — cake dimensions; a_x, a_y — x and y coordinates of the first candle; b_x, b_y — the coordinates of the second candle ($3 \leq w, h \leq 10^9$; $0 < a_x, b_x < w$; $0 < a_y, b_y < h$; $a_x \neq b_x$ or $a_y \neq b_y$).

Output

Output four integers $s_x, s_y, e_x,$ and e_y — the starting and ending coordinates of the cut. Both starting and ending point of the cut should belong to the sides of the cake.

If there are several solutions, output any of them.

Example

anniversary.in	anniversary.out
7 3 2 2 3 2	0 0 4 3

Problem B. Boys and Girls

Input file: boysgirls.in
Output file: boysgirls.out
Time limit: 2 seconds
Memory limit: 256 megabytes

Bob found a nice task in his old math book for children. It says:

There are 10 children standing in a circle, 5 of them stand next to a boy, and 7 of them stand next to a girl. How is it possible?

Here is the solution to the task. If 4 boys and 6 girls stand like this: BGBGBGBGBG, there are 5 children who stand next to a boy (here they are underlined: BGBGBGBGBG), and 7 children who stand next to a girl (BGBGBGBGBG).

Now Bob wants to solve a generalized version of this task:

There are n children standing in a circle, x of them stand next to a boy, and y of them stand next to a girl. How is it possible?

Help Bob by writing a program that solves the generalized task.

Input

The single line of the input contains three integers n , x and y ($2 \leq n \leq 100\,000$; $0 \leq x, y \leq n$).

Output

If there is a solution, output a string of length n , describing the order of children in the circle. Character 'G' corresponds to a girl, character 'B' corresponds to a boy. If there are several solutions, output any of them.

If there is no solution, output "Impossible".

Examples

boysgirls.in	boysgirls.out
10 5 7	BGBGBGBGGG
10 3 8	Impossible

Problem C. CodeCoder vs TopForces

Input file: codecoder.in
Output file: codecoder.out
Time limit: 2 seconds
Memory limit: 256 megabytes

Competitive programming is very popular in Byteland. In fact, every Bytelandian citizen is registered at two programming sites — CodeCoder and TopForces. Each site maintains its own proprietary rating system. Each citizen has a unique integer rating at each site that approximates their skill. Greater rating corresponds to better skill.

People of Byteland are naturally optimistic. Citizen A thinks that he has a chance to beat citizen B in a programming competition if there exists a sequence of Bytelandian citizens $A = P_0, P_1, \dots, P_k = B$ for some $k \geq 1$ such that for each i ($0 \leq i < k$), P_i has higher rating than P_{i+1} at one or both sites.

Each Bytelandian citizen wants to know how many other citizens they can possibly beat in a programming competition.

Input

The first line of the input contains an integer n — the number of citizens ($1 \leq n \leq 100\,000$). The following n lines contain information about ratings. The i -th of them contains two integers CC_i and TF_i — ratings of the i -th citizen at CodeCoder and TopForces ($1 \leq CC_i, TF_i \leq 10^6$). All the ratings at each site are distinct.

Output

For each citizen i output an integer b_i — how many other citizens they can possibly beat in a programming competition. Each b_i should be printed in a separate line, in the order the citizens are given in the input.

Example

codecoder.in	codecoder.out
4	2
2 3	2
3 2	0
1 1	3
4 5	

Problem D. Digital Addition

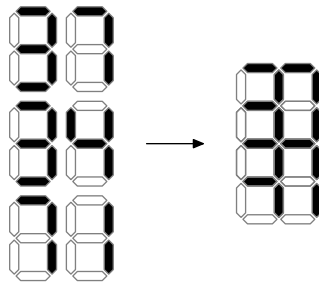
Input file: `digital.in`
Output file: `digital.out`
Time limit: 2 seconds
Memory limit: 256 megabytes

Deidra is doing a columnar addition. She writes down two non-negative integer summands one below the other, left-pads them with zeroes so that they have equal length, and calculates the sum (e. g. $77 + 05 = 82$). If the sum is longer than each of the summands (because of a carry, as in $96 + 07 = 103$) she appends a zero at the beginning of each summand ($096 + 007 = 103$). She allows herself to use unnecessary leading zeroes ($007 + 004 = 011$) as soon as the length of all three numbers is the same.

Also Deidra has a homemade printing press. She decided to print her addition without a plus or a horizontal line, using the following standard font:



Unfortunately, she messed up with spacing, and all the digits were printed over each other in the following way. Digits that were supposed to be horizontally adjacent were printed so that the right two segments of the left digit coincide with the left two segments of the right digit. Digits that were supposed to be vertically adjacent were printed so that the bottom half (a square with 4 segments) of the upper digit coincides with the top half of the lower digit.



When one or more black segments are printed at the same position, the result looks black. When only empty segments are printed at the position, the result looks white.

Given the resulting picture, find a correct addition that could produce it or detect that there is none.

Input

The first line of the input contains an integer w ($1 \leq w \leq 100$) — the width of Deidra's addition (number of digits in each line).

The following 9 lines contain the description of the picture printed with the bad spacing. Each line contains w or $w + 1$ digits. '1' denotes a black segment, '0' denotes a white one. Even lines start with a space. See the examples for clarification.

Output

If there is no valid addition, output "NO".

Otherwise output a valid addition that produces the given figure. The output should consist of three lines, each containing w digits.

If there are several solutions, output any of them.

Examples

digital.in	digital.out
2 1 1 0 1 1 1 0 0 1 1 1 1 0 1 1 1 0 0 1 1 0 0	37 34 71
1 1 0 1 1 1 1 1 1 1 1 0 1 0	2 2 4
1 1 1 0 1 1 1 1 1 1 1 0 1 0	NO

The first example is illustrated above in the problem statement.

In the second example an addition $7 + 0 = 7$ also produces the same picture.

Problem E. Easy Reading

Input file: `easy.in`
Output file: `easy.out`
Time limit: 2 seconds
Memory limit: 256 megabytes

Eugene is reading a boring book. To make this process interesting he is drawing a picture at the same time. He has a piece of graph paper that is divided into square cells. All the cells are empty at the beginning.

Eugene starts by painting over one cell. Then he opens the book at a random page and starts reading. Whenever he sees the letter ‘u’ in the text, he moves his pen one cell up and then paints over the cell under his pen. Whenever he sees the letter ‘d’, he does the same but moves his pen one cell down instead of up. For letters ‘l’ and ‘r’ he moves to the left and to the right, respectively. If he wants to paint a cell that was already painted, he paints it again.

You have found a piece of paper and the text of the book. Now you want to understand if the picture on the paper could be drawn by Eugene at some point of his book reading. Remember that Eugene could use some substring of the text.

Input

The first line of the input contains an integer l — the length of the text ($1 \leq l \leq 100\,000$). The second line contains a string of length l — the text. It contains only lowercase English letters, spaces, commas and periods. The text neither begins nor ends with a space.

The third line contains two positive integers n and m — the picture dimensions ($2 \leq n \times m \leq 100\,000$).

Each of the following n lines contains a string of length m . Painted cells are denoted by ‘X’, while empty cells — by ‘.’. It is guaranteed that there are at least two painted cells in the picture.

The first of these n lines corresponds to the top of the picture and the last one corresponds to the bottom of it.

Output

If the picture could be painted by Eugene, output “YES” on the first line. On the second line print two integers b and e such that if Eugene read all letters from b to e , inclusive, he would draw exactly the same picture as described in the input ($1 \leq b \leq e \leq l$). If there are several solutions, output any of them.

If the picture couldn’t be drawn by Eugene, output “NO”.

Examples

<code>easy.in</code>	<code>easy.out</code>
43 you should read statement really carefully. 3 6 ...XX. ..XXX. ...XXX	YES 3 42
43 you should read statement really carefully. 3 2 XX XX XX	NO

Problem F. Folding

Input file: `folding.in`
Output file: `folding.out`
Time limit: 2 seconds
Memory limit: 256 megabytes

As you can remember, Alex is fond of origami. She switched from squares to rectangles, and rectangles are much more difficult to master. Her main interest is to determine what is the minimum possible number of folds required to transform $W \times H$ rectangle to $w \times h$ one. The result of each fold should also be rectangular, so it is only allowed to make folds that are parallel to the sides of the rectangle.

Help Alex and write a program that determines the minimum required number of folds.

Input

The first line of the input contains two integers W and H — the initial rectangle dimensions. The second line contains two more integers w and h — the target rectangle dimensions ($1 \leq W, H, w, h \leq 10^9$).

Output

Output a single integer — the minimum required number of folds to transform the initial rectangle to the target one.

If the required transformation is not possible, output -1 .

Examples

<code>folding.in</code>	<code>folding.out</code>
2 7 2 2	2
10 6 4 8	2
5 5 1 6	-1

In the first example you should fold 2×7 rectangle to 2×4 , and then to 2×2 .

In the second example you should fold 10×6 rectangle to 10×4 , then to 8×4 , and rotate it to 4×8 .

In the third example it is impossible to fold 5×5 rectangle to 1×6 one (remember that folds must be parallel to the rectangle sides).

Problem G. Gangsters in Central City

Input file: `gangsters.in`
Output file: `gangsters.out`
Time limit: 2 seconds
Memory limit: 256 megabytes

For a long time, there were no problems with water in Central City. The sewage of the city has a form of a rooted tree: the central reservoir is situated at the root and the houses are at the leaves. The water flows from the central reservoir to the houses by the pipes that runs along the edges of the tree. Each house has an access to water.

Suddenly, gangsters captured some of the houses. As a mayor of the city you are very concerned, and you want to kick out the gangsters. So you want to stop the water flow to houses captured by the gangsters. To do that you could clog some pipes of the sewage system. If the path from the reservoir to a house contains at least one clogged pipe, the house does not have an access to water.

You are very afraid of the gangsters, so you decided to clog up the minimal number of pipes, that it could look like an accident. At the same time, you care about the citizens, so for the chosen number of clogged pipes, you want to minimize the number of houses without gangsters and access to water.

Unfortunately, the gangsters could appear and disappear from some houses. So, you are asking the scientists about the minimum required number of clogged pipes and the minimum required number of houses without gangsters and access to water after each change in the gangsters' location.

Input

The first line of the input contains two integers n and q — the number of vertices in the tree which represents the sewage and the number of changes in the location of the gangsters ($2 \leq n \leq 100\,000$; $1 \leq q \leq 100\,000$).

The second line contains the description of the sewage: a sequence of $n - 1$ integers p_2, p_3, \dots, p_n , where p_i is the parent of the vertex i ($1 \leq p_i < i$). The central reservoir is located at the vertex 1.

The next q lines represent the changes in the location of the gangsters. Each change could be one of two different types: “+ v ” — the gangsters capture the house at vertex v ; “- v ” — the gangsters leave the house at vertex v .

At the beginning all the houses are free of gangsters. All the changes form the correct sequence: the gangsters cannot capture the house if it is already captured and the gangsters could not leave the house if it is not captured.

Output

The output should contain $2q$ integers, two in each line: c_i — the minimum number of clogged pipes and h_i — the minimum number of houses without gangsters and have no access to water for the chosen c_i .

Example

<code>gangsters.in</code>	<code>gangsters.out</code>
7 6	1 0
1 2 1 3 3 3	2 0
+ 4	2 1
+ 5	2 0
+ 6	2 1
+ 7	2 0
- 6	
- 5	

Problem H. Hard Cuts

Input file: `hard.in`
 Output file: `hard.out`
 Time limit: 2 seconds
 Memory limit: 256 megabytes

Given a rectangle with integer side lengths, your task is to cut it into the smallest possible number of squares with integer side lengths.

Input

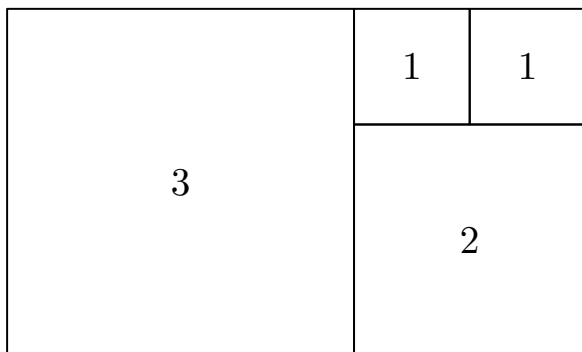
The first line contains a single integer T — the number of test cases ($1 \leq T \leq 3600$). Each of the next T lines contains two integers w_i, h_i — the dimensions of the rectangle ($1 \leq w_i, h_i \leq 60$; for any $i \neq j$, either $w_i \neq w_j$ or $h_i \neq h_j$).

Output

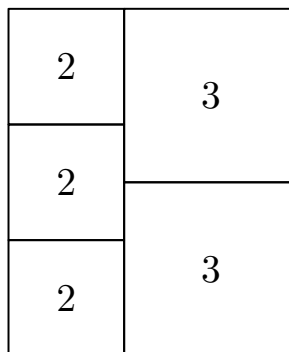
For the i -th test case, output k_i — the minimal number of squares, such that it is possible to cut the w_i by h_i rectangle into k_i squares. The following k_i lines should contain three integers each: x_{ij}, y_{ij} — the coordinates of the bottom-left corner of the j -th square and l_{ij} — its side length ($0 \leq x_{ij} \leq w_i - l_{ij}$; $0 \leq y_{ij} \leq h_i - l_{ij}$). The bottom-left corner of the rectangle has coordinates $(0, 0)$ and the top-right corner has coordinates (w_i, h_i) .

Example

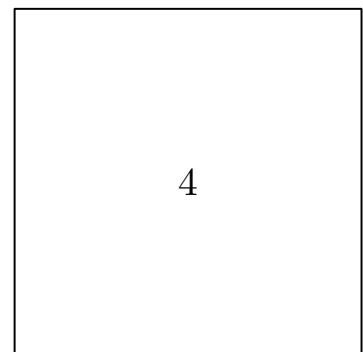
hard.in	hard.out
3	4
5 3	0 0 3
5 6	3 0 2
4 4	3 2 1
	4 2 1
	5
	0 0 2
	0 2 2
	0 4 2
	2 0 3
	2 3 3
	1
	0 0 4



Example case 1



Example case 2



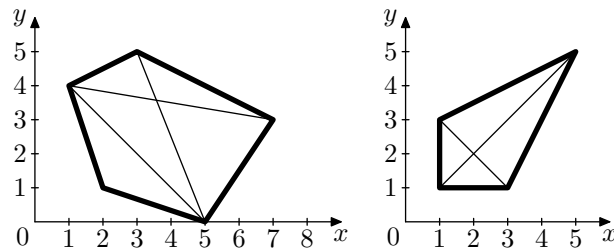
Example case 3

Problem I. Integral Polygons

Input file: `integral.in`
 Output file: `integral.out`
 Time limit: 2 seconds
 Memory limit: 256 megabytes

Ingrid holds a polygon shop in a far away country. She sells only convex polygons with integer coordinates. Her customers prefer polygons that can be cut into two halves *in a proper way*, that is the cut should be straight with starting and ending points in the polygon vertices and both halves should be non-empty and have integer areas. The more ways to cut the polygon in the proper way are — the more expensive the polygon is.

For example, there are three ways to cut the left polygon in the proper way, and two ways for the right polygon.



The polygons in the shop are always of excellent quality, so the business is expanding. Now Ingrid needs some automatic tool to determine the number of ways to cut the polygon in the proper way. This is very important for her shop, since otherwise you will spend a lot of time on setting prices — just imagine how much time would it take to set prices for a medium-sized van with polygons. Could you help Ingrid and write the tool for her?

Input

The first line of the input contains an integer n — the number of polygon vertices ($4 \leq n \leq 200\,000$). Each of the following n lines contains vertex coordinates: a pair of integers x_i and y_i per line ($-10^9 \leq x_i, y_i \leq 10^9$).

The specified polygon is convex and its vertices are specified in the order of traversal.

Output

Output a single integer w — the number of ways to cut the polygon in the proper way.

Example

<code>integral.in</code>	<code>integral.out</code>
5 7 3 3 5 1 4 2 1 5 0	3
4 1 1 3 1 5 5 1 3	2

Problem J. Java2016

Input file: java2016.in
 Output file: java2016.out
 Time limit: 2 seconds
 Memory limit: 256 megabytes

John likes to learn esoteric programming languages. Recently he discovered the probabilistic programming language Java2K. Built-in functions of Java2K have only a certain probability to do whatever you intend them to do.

The Java2K programming is very hard, so John designed a much simpler language for training: Java2016. Built-in operators of Java2016 are deterministic, while their operands are random. Each value in Java2016 is a positive integer in the range 0..255, inclusive.

Java2016 supports six operators of three precedencies:

$$\begin{aligned} \langle \text{expression} \rangle &::= \langle \text{expression} \rangle \text{'min'} \langle \text{sum} \rangle \mid \langle \text{expression} \rangle \text{'max'} \langle \text{sum} \rangle \mid \langle \text{sum} \rangle \\ \langle \text{sum} \rangle &::= \langle \text{sum} \rangle \text{'+'} \langle \text{term} \rangle \mid \langle \text{sum} \rangle \text{'-'} \langle \text{term} \rangle \mid \langle \text{term} \rangle \\ \langle \text{term} \rangle &::= \langle \text{term} \rangle \text{'*'} \langle \text{factor} \rangle \mid \langle \text{term} \rangle \text{'/'} \langle \text{factor} \rangle \mid \langle \text{factor} \rangle \\ \langle \text{factor} \rangle &::= \text{'('} \langle \text{expression} \rangle \text{'')} \mid \text{'?' } \mid \langle \text{macro} \rangle \end{aligned}$$

Minimum ('min') and maximum ('max') operators are defined as usual. Addition ('+'), subtraction ('-') and multiplication ('*') are defined modulo 256. The result of the division ('/') is rounded towards zero. If the divider is zero, the program crashes. The argument of the operator is a result of another operator, evenly distributed random value ('?'), or macro substitution.

For instance, the probability that "?/?/?" is evaluated to zero is 98.2%, while the probability of the crash is 0.8%.

The Java2016 program consists of zero or more macro definitions, followed by the resulting expression. Each macro definition has a form of

$$\begin{aligned} \langle \text{macrodef} \rangle &::= \langle \text{macro} \rangle \text{'='} \langle \text{expression} \rangle \\ \langle \text{macro} \rangle &::= \text{'a'... 'z'} \end{aligned}$$

The macro should be defined before the first use. It may not be redefined. The macro is expanded to its definition on each use. For instance,

```
a = ? max ?
(a max a) / a
```

is expanded to "((? max ?) max (? max ?)) / (? max ?)".

John is going to add probabilistic constants to Java2016, so for each possible constant value he needs a program that successfully evaluates to this value with at least one-half probability. Crashes are counted toward failures.

Input

The input contains a single integer c — the target constant ($0 \leq c \leq 255$).

Output

Output a Java2016 program that successfully evaluates to constant c with probability no less than $1/2$. The total length of the program should not exceed 256 characters (excluding spaces).

Examples

java2016.in	java2016.out
0	? /?/ ?
1	a = ? max ? (a max a) / a

Problem K. King's Heir

Input file: king.in
Output file: king.out
Time limit: 2 seconds
Memory limit: 256 megabytes

The king is dead, long live the king! After the sudden death of the king Fert XIII the people of the Flatland Kingdom are going to welcome the new king. Unfortunately, there is a problem, Fert has too many sons.

Actually, he has n sons and he loved each new son more than all of his previous sons. Well, probably he just stopped loving his sons because of their bad behavior. Anyway, after the new son was born Fert made the new testament that declared that the newly born son would be the heir.

However, there is a problem. Only the king's son who is at least 18 years old at the moment of the king's death can become a new king. Now the ministers of the government are trying to find the correct new king, but they seem to fail. Help them!

Input

The first line of the input contains three integers: d , m and y — the day, the month and the year of the king's death, d is from 1 to 31, m is from 1 to 12, y is from 1 to 9999. It is guaranteed that there exists day d in month m , all months have the same number of days in Flatland as in our country, except that Flatland calendar doesn't have leap years, so February (month 2) always has 28 days.

The second line contains n ($1 \leq n \leq 100$) — the number of king's sons. The following n lines contain three integers each d_i , m_i and y_i and specify the birth dates of king's sons. All dates are correct and no son is born after or on the day of king's death. The king had no twins, so no two sons were born on the same date.

Output

Output one integer — the number of the son that would become the king, or -1 if none of them is at least 18 years old. The sons are numbered from 1 to n in order they are described in the input. The youngest son who is at least 18 years old at the moment of the king's death would become the king. If the son has his 18th birthday exactly on the day of the king's death, he can become a king.

Examples

king.in	king.out
22 10 2016 7 28 2 1999 22 7 1995 21 10 1998 23 10 1998 3 9 2000 1 4 2013 17 12 2004	3
22 10 2016 1 28 2 1999	-1