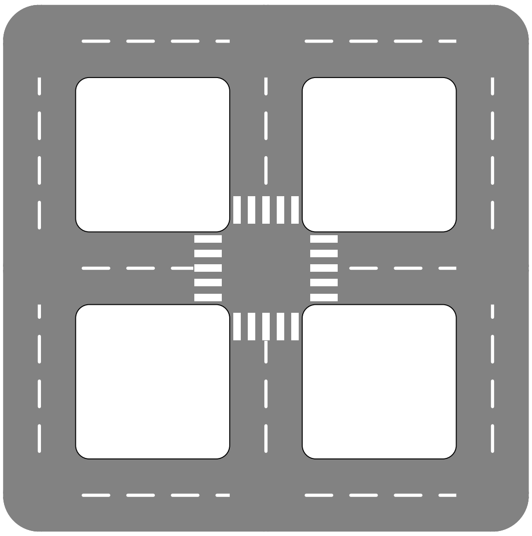
Problem A. Crossings

Input file: `standard input`
Output file: `standard output`
Time limit: 1 second
Memory limit: 256 megabytes

You are designing your dream planned city built around automated cars. The city consists of a perfect grid of $2^n \times 2^n$ square blocks, with straight streets between them. You'd want your city to have nothing impeding traffic. Unfortunately, local laws require that your city is traversable by pedestrians – one should be able to walk from any block to any other block.

You therefore need to slow down traffic by putting pedestrian crossings at some intersections. You can choose an intersection and add to it four zebra crossings, which allow people to walk between any of the four adjacent blocks. From those, they can reach any other crossing adjacent to those blocks and so on. Determine the minimum number of such intersections required so that pedestrians can walk from any block to any other block.



Input

The input consists of a single integer n ($1 \leq n \leq 30$).

Output

Output a single integer, the minimum number of intersections with pedestrian crossings required.

Examples

standard input	standard output
1	1
26	1501199875790165

Problem B. Binary Game

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Alice and Bob are playing a game in which they have to write down a string that consists of exactly n zeros and m ones. The players alternate their moves, and Alice goes first. In a single move, a player can append a binary digit to either the front or the end of the current string; initially, the string is empty. The string is not allowed to contain more than n zeros or more than m ones at any point.

At the end of the game, the players examine the string they have obtained as an integer in binary. Alice's goal is to minimize the resulting number, and Bob's goal is to maximize it. Given that both players play optimally, find the final binary string they will obtain in the game!

Input

The input contains two integers n and m ($1 \le n, m \le 10^5$), the number of 0s and 1s in the final string.

Output

Output a single string containing n zeros and m ones, the final result the players will obtain if both play optimally.

Example

standard input	standard output
1 3	1101

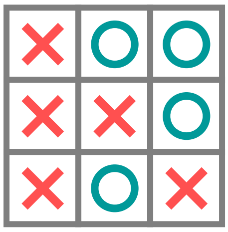
Problem C. Tic-Tac-Toe

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

In Generalized Tic-Tac-Toe, two players play on an $n \times n$ cell grid. The rules of the game are as follows: initially, the grid is empty, and the players alternate turns. The first player in a single move draws a cross in an empty cell; the second player in a single move draws a circle in an empty cell. If at any point there are 3 equal symbols in consecutive horizontal, vertical, or diagonal cells, the corresponding player wins, and the game ends. If all cells are filled and no player has won, the game ends in a tie.

You are given an integer m, the number of moves in an instance of Generalized Tic-Tac-Toe that did not end in a tie. Your task is to find any valid game protocol with exactly m moves such that the game ends with a player winning on the m-th move and at no previous point does the winning condition hold.

For example, consider a case with $n = 3$ and $m = 9$, and suppose that the final grid is equal to the one shown in the figure. Then only game sequences such that the last move is made in the upper left corner will be valid; otherwise, the winning condition would have held for the first player before the 9th move.



Input

The first line contains two integers, n ($3 \le n \le 100$) and m ($5 \le m \le n^2$).

Output

Output m lines where the k-th line contains two integers i and j, the line and column indices of the cell filled in the k-th move. If there are multiple solutions, output any of those. It can be proven that a solution always exists.

Examples

standard input	standard output
3 9	2 1 1 3 2 2 2 3 3 3 3 2 3 1 1 2 1 1
4 6	1 1 4 4 1 2 4 3 1 4 4 2

Problem D. Dark Web

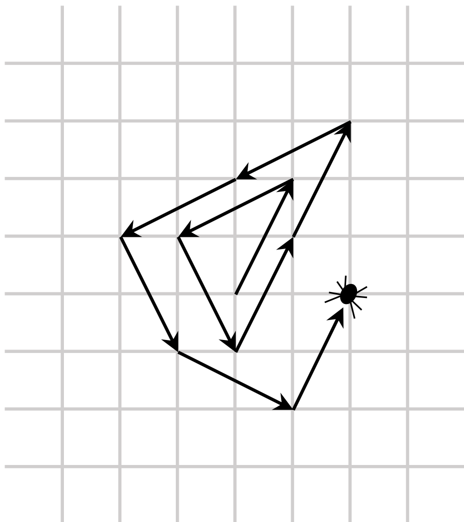
Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

There are many studies on the effects of psychoactive drugs on animals. This time, the scientists from the Experimental University decided to test how amphetamine impacts spider web weaving.

The results were intriguing: first of all, the web strings became a much darker color. More importantly for us, it appears that spiders start to construct webs of a meticulously obsessive pattern. The scientists were even able to find a precise mathematical model describing such webs.

We examine a single spider's position in a 2-dimensional plane with integer coordinates. Initially, the spider is located at the point $(0, 0)$ and it starts weaving its web from there. In a single move, the spider will move by a vector (a_x, a_y), producing a straight segment of web connecting the previous and the next point. This segment will satisfy either $|a_x| = 1$ and $|a_y| = 2$, or $|a_x| = 2$ and $|a_y| = 1$. The spider always chooses such a vector so that it has no common points with any previous segment, except for the last point of the last segment. Additionally, the clockwise angle between the previous segment and the next must be the smallest possible.

In the first move, the spider moves by the vector $(1, 2)$. It then follows the rules above. As a result, the web resembles a dark spiral woven in the counter-clockwise direction, as seen in the figure below.



The scientists want to be able to predict the behavior of the spider based on this model. Determine the coordinates of the spider's location after n moves!

Input

The input contains a single integer n ($1 \le n \le 10^6$), the number of moves.

Output

Output two integers x and y, the coordinates of the spider after n moves.

Examples

standard input	standard output
`10`	`2 0`
`42`	`1 -5`

Problem E. Scorpion

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	256 megabytes

You are a train driver and are trying to move a train out of the otherwise empty Scorpion rail yard. The rail yard consists of n junctions connected by $n-1$ rail segments of various lengths. The junctions are numbered from 1 to n. Every junction is reachable from any other junction.

The front of the train is currently at junction a, and the rear is at junction b, with the train occupying the rails between them. The train can move in either direction while remaining a fixed length, and can turn onto any connected rail when at a junction. Note that, while they start that way, the ends of the train don't necessarily have to be at junctions.

In order to exit the rail yard, you need to get the front of the train to junction t. Calculate the minimum total distance you need to move the train to get the front of it there.

Input

The first line contains a single integer n ($2 \le n \le 10^6$), the number of junctions in the rail yard. The second line contains two integers a, b ($1 \le a, b \le n$), the starting positions of the front and the rear of the train. It is guaranteed that $a \ne b$. The third line contains a single integer, the exit junction t ($1 \le t \le n$). The following $n-1$ lines each contains three integers a_i, b_i, l_i ($1 \le a_i, b_i \le n$, $1 \le l_i \le 10^6$) describing the two end junctions and length of the i-th rail segment. It is guaranteed that all the junctions in the rail yard are connected.

Output

Output a single integer, the minimum total distance the train needs to travel to reach the exit junction with its front or -1 if this is impossible.

Examples

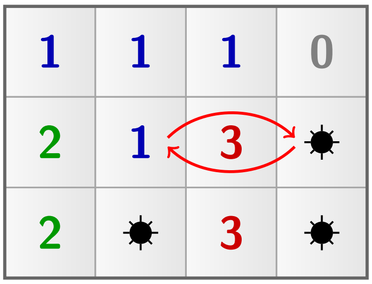
standard input	standard output
5 1 3 5 1 2 1 2 3 10 3 4 8 2 5 3	4
3 1 2 3 1 2 1 2 3 1	-1

Problem F. Mineswapper

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

You are learning a new game, Mineswapper. In this game, you are given an $n \times m$ grid of cells. Each cell contains either a mine, or a digit between 0 and 8.

You can swap the location of a mine and a digit. You are currently playing on the easiest difficulty, where you can do this at most once. Your goal is to obtain a grid where each cell with a digit has exactly as many mines in all of its neighboring cells (of which there are at most eight) as the value of this digit.



If the grid is already in this state, you should make no moves. It is guaranteed that at most one move is necessary to reach the target state.

Input

The first line contains two integers, n ($1 \leq n \leq 500$) and m ($1 \leq m \leq 500$). The next n lines contain m symbols each: the contents of the game grid. Each symbol is either a '`*`', corresponding to a mine, or a digit from 0 to 8.

It is guaranteed that at most one move is necessary to reach the target state from the given grid.

Output

In the first line output either 0 or 1, the minimum number of required moves.

If this number is 1, the next two lines should each contain a pair of integers r_1, c_1 and r_2, c_2 ($1 \leq r_1, r_2 \leq n$, $1 \leq c_1, c_2 \leq m$), such that the single move swaps the contents of the cell in row r_1, column c_1 with the one in row r_2, column c_2, with the rows numbered from top to bottom. Exactly one of these cells should contain a mine.

It there exist multiple correct solutions, output any of them.

Examples

standard input	standard output

`3 4`
`1110`
`213*`
`2*3*`
	`1`
	`2 2`
	`2 4`

`3 5`
`*3***`
`14*8*`
`02***`
	`0`

Problem G. Curling Golf

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

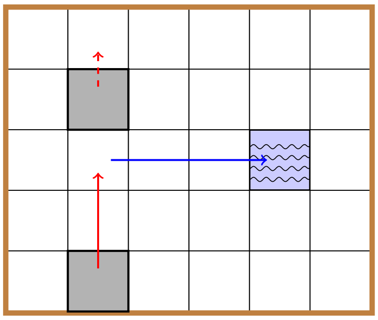
Curling Golf is played on a $w \times h$ grid of ice tiles over a frozen lake bordered by a fence. Two of the tiles contain a square curling stone. One of the tiles has a hole cut into the ice.

In a single move, you may slide either of the stones in any of the four orthogonal directions. The stone continues sliding in this direction until it encounters an obstacle:

- if it hits a fence, it stops;
- if it hits the other stone, it stops on the tile before the other stone, and that stone starts sliding in the same direction instead;
- if it hits the hole, it sinks.

After both stones have stopped, you may make another move, again using either of the stones. The goal of the game is to sink either one of the stones in as few moves as possible.

In the following illustration, the first move is colored red, and slides the other stone as well. The second move is colored blue.



Input

The first line contains two integers w, h ($1 \leq w, h \leq 10^9$), the size of the grid. The i-th of the next two lines each contains a pair of coordinates x_i, y_i ($1 \leq x_i \leq w$, $1 \leq y_i \leq h$), the starting location of one of the stones. Finally the last line contains a pair of coordinates x_h, y_h ($1 \leq x_h \leq w$, $1 \leq y_h \leq h$), the location of the hole.

It is guaranteed that all three pairs of coordinates are different.

Output

Output a single integer, the minimum number of moves required to sink one of the stones. It can be proven that it is always possible to do so.

Example

standard input	standard output
6 5 2 1 2 4 5 3	2

Problem H. Cell Towers

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

As a manager of the popular telecommunications provider Tele1, you are in charge of setting up cell towers in a 1-dimensional city which consists of a segment of n cells. A tower built in the i-th cell would have coverage of all cells in the interval $[i - r_i, i + r_i]$. Your task is to find the smallest number of towers to build so that all cells of the city are covered by the communications of at least one cell tower.

Input

The first line contains an integer n ($1 \leq n \leq 2 \cdot 10^5$), the number of cells of the city. The next line contains n integers $r_1, \ldots, r_n$ ($0 \leq r_i < n$), the coverage radii of the possible towers.

Output

Output a single number, the smallest number of cell towers to build to cover the whole city with telecommunications.

Examples

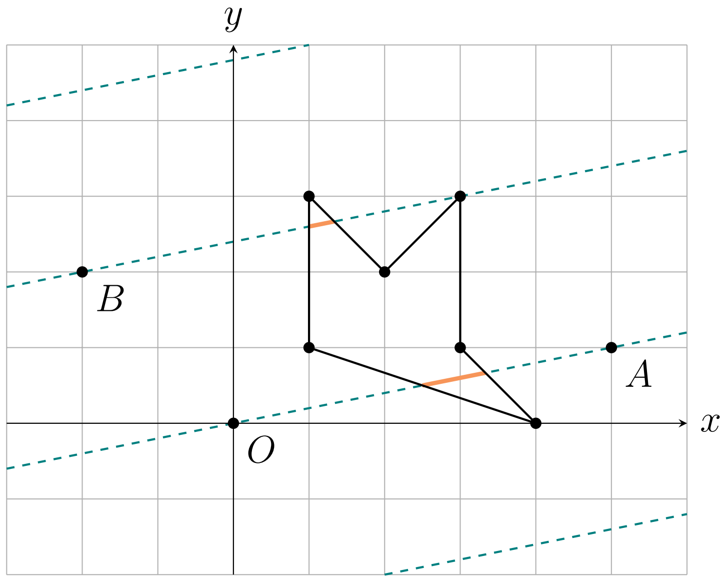
standard input	standard output
9 3 1 0 2 2 0 0 0 1	3
5 4 2 0 1 3	1

Problem I. Polygon

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

You are given a polygon with n vertices and an infinite set of equally spaced parallel lines called *stripes*. Your task is to calculate the total length of stripe segments that are inside or on the edges of the polygon.

For example, in the following figure, the stripes are dashed, and their interior segments in the polygon are highlighted solid.



Input

The first line contains an integer t ($1 \le t \le 80$), the number of test cases. The first line of each test case contains n ($3 \le n \le 10^6$), the number of points of the polygon. The second line contains two integers $-1\,000 \le A_x, A_y \le 1\,000$, and the third line contains two integers $-1\,000 \le B_x, B_y \le 1\,000$. This means that one stripe passes through the points $O = (0, 0)$ and $A = (A_x, A_y)$ and another stripe that is parallel to the first one passes through $B = (B_x, B_y)$. These two stripes define the spacing: there is no other stripe in between them. It is guaranteed that all three points O, A and B are distinct.

The i-th of the next n lines contains two integers x_i, y_i ($-5 \cdot 10^5 \le x_i, y_i \le 5 \cdot 10^5$), the coordinates of the i-th polygon vertex. It is guaranteed that all the vertices are distinct, the polygon doesn't intersect itself (even at a point) and no three consecutive vertices are collinear. The vertices are given in the order they appear on the perimeter, but this can be either clockwise or counterclockwise.

The total sum of n over all t test cases is at most 10^6.

Output

Output a single real number, the total length of the segments of the stripes inside or on the edges of the polygon. The output will be considered correct if the absolute or relative error doesn't exceed 10^{-4}.

Examples

standard input	standard output
1 6 5 1 -2 2 1 1 1 3 2 2 3 3 3 1 4 0	1.189771
2 4 0 2 1 0 1 1 1 3 3 3 3 1 4 -1 1 0 -1 1 1 1 3 3 3 3 1	6.000000 5.656854

Problem J. Reductionism

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

Suppose you have two positive integers a and b written on a whiteboard. In a single move, you can write down another integer at the end of the sequence so that if the last three numbers in the sequence are x, y, z, then the product of two of them must be equal to the third one. If it is possible to obtain the integer 1 at the end of the sequence, we call the starting pair (a, b) *reductible* (regardless of the length of the sequence).

For example, the pair $(4, 8)$ is reductible, such as via the following sequence:

$$4, 8, 2, 4, 2, 2, 1.$$

Given n positive integers a_1, $\ldots$, a_n, determine the total number of reductible pairs.

Input

The first line contains a single integer n ($2 \le n \le 10^6$). Then follow n integers a_1, $\ldots$, a_n ($2 \le a_i \le 10^6$).

Output

Output a single number, the total number of reductible pairs.

Examples

standard input	standard output
3 2 3 4	1
4 4 6 4 8	3

Problem K. Implications

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

In Boolean logic, the values True and False are usually denoted by 1 and 0, respectively. The *implication* operator $\implies$ is a relation between two Boolean values, itself takes a Boolean value and is defined as follows:

a	b	$a \implies b$
0	0	1
0	1	1
1	0	0
1	1	1

In other words, anything follows from a false statement, but only truth follows from truth. You have recovered an ancient scroll containing the string $x_1 x_2 \ldots x_n$, where each symbol is either `0`, `1` or `?`. Your task is to determine any possible way to substitute the `?` symbols with the values `0` and `1` so that the formula

$$x_1 \implies x_2 \implies \ldots \implies x_n$$

is equal to 1, or determine that it is impossible. Here, the formula is evaluated right-to-left, which means that for an expression $a \implies b \implies c$, the value $d = (b \implies c)$ is evaluated first and only then is $a \implies d$ evaluated.

Input

The first line contains an integer n ($1 \leq n \leq 10^6$), the length of the string. The next line contains a string of length n containing only the symbols `0`, `1` and `?`.

Output

If it is possible, output "`Yes`" in the first line. In the next line, output a string $y_1 y_2 \ldots y_n$ consisting only of `0`s and `1`s such that for all i, if $x_i \neq$ `?`, then $y_i = x_i$. The given formula must evaluate to 1 on this string. If there are multiple solutions, output any of them.

If the required is not possible, output "`No`".

Examples

standard input	standard output
4 1??1	Yes 1101
2 10	No

Problem L. Badness 0

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

After decades of struggle, you have finally single-handedly proved the famous Goldbach's conjecture. Naturally, you wish your paper to be as beautiful as possible. For instance, you want your text to be perfectly evenly spaced, something that existing typesetting systems like LaTeX do not support.

Therefore, not unlike some of the greats, you are now developing your own prototype of such a system. You have decided on the following requirements for it:

- the output should be in a monospaced font, where each symbol (including spaces) has the exact same width;
- each token of your text should be placed wholly in a single line, never split across lines;
- there should also be a single space between each two consecutive tokens in a line;
- the text shouldn't contain other characters;
- the width of your text should be as small as possible because it makes the text easier to read.

For example, here is your (currently manually) perfectly aligned abstract in two lines:

```
We prove that each even integer greater than
two is expressible as the sum of two primes.
```

As you can see, the length of each line is exactly the same. Given your manuscript as a list of tokens, write a program that produces a text that is perfectly evenly aligned and has the smallest width!

Input

The first line contains an integer n ($1 \le n \le 1\,000$), the number of tokens. Then follow n strings, the tokens of your text. Each two consecutive strings are separated by either a space or a new line. The length of each string is at most 50 characters. Each token consists only of lower and upper case English alphabet letters, digits, and the punctuation marks '`,`', '`.`', '`:`', '`;`', '`-`', '`!`', '`?`'.

Output

First output an integer w, the minimum width of your text. Then typeset the given text, w characters in each line, with every two consecutive tokens in each line separated by a single space.

Examples

standard input	standard output
19 Lorem ipsum dolor sit amet, consetur adipisci elit. Sed do modum tempos incidunt ut labore et dolore magna est.	27 Lorem ipsum dolor sit amet, consetur adipisci elit. Sed do modum tempos incidunt ut labore et dolore magna est.
3 Quod erat demonstrandum.	24 Quod erat demonstrandum.