TASK	IPv6	LOGO	PLINOVOD
input	standard input		
output	standard output		
time limit (per test case)	1 second		
memory limit (heap+stack)	64 MB		
points	**40**	**50**	**60**
	150		

In IPv6, the next generation of the Internet protocol, an IP address is 128 bits long. The full text representation of an IPv6 address consists of 8 groups of 4 hexadecimal digits each, where groups are separated by colon characters ':'. For example:

```
2001:0db8:85a3:0000:0000:8a2e:0370:7334
```

To compress this textual representation, we are allowed the following transformations:

- All or some **leading zeroes** in a group can be removed. For example, the above address can be simplified as:

  ```
  2001:db8:85a3:0:00:8a2e:370:7334
  ```

- Additionally, one or more **consecutive groups of zeroes** can be replaced by a double colon "::". The above address becomes:

  ```
  2001:db8:85a3::8a2e:370:7334
  ```

  This last simplification can be done **only once** so that the resulting text representation is never ambiguous.

Write a program that takes a valid (possibly compressed) text representation of an IPv6 address and outputs its full text representation.

INPUT

The first line contains a string of at most 39 characters, a valid text representation of an IPv6 address. The string consists of digits '0'-'9', lowercase letters 'a'-'f' and colon characters ':'.

OUTPUT

Output the full representation of the given IPv6 address.

EXAMPLES

input	**input**
`25:09:1985:aa:091:4846:374:bb`	`::1`
output	**output**
`0025:0009:1985:00aa:0091:4846:0374:00bb`	`0000:0000:0000:0000:0000:0000:0000:0001`

Competition tasks for the programming language LOGO often involve drawing rectangles on the screen. Drawing in LOGO is done by moving a turtle-shaped cursor.

The turtle is defined by its position and angle. It holds in its mouth a pencil that can be raised or lowered. When the pencil is lowered, moving the turtle causes the pencil to leave a trail on the screen.

The turtle is initially at coordinates (0, 0) facing in the direction of increasing y coordinates, pencil lowered. We can manipulate the turtle with the following set of commands:

1. FD x – move the turtle x units forward.
2. LT α – turn the turtle α degrees counter-clockwise.
3. RT α – turn the turtle α degrees clockwise.
4. PU – raise the pencil.
5. PD – lower the pencil.

Your program will be given a set of rectangles (with sides parallel to coordinate axes) that need to be drawn on the screen. The turtle may go over the same segment with its pencil lowered multiple times, but it is not allowed to draw anything other than the given rectangles.

Write a program that calculates the **smallest** number of times the pencil must be **raised** in order to draw all the rectangles.

INPUT

The first line contains an integer N ($1 \leq N \leq 1000$), the number of rectangles.

Each of the following N lines contains for integers x_1, y_1, x_2 and y_2 ($-500 \leq x_1 < x_2 \leq 500$), ($-500 \leq y_1 < y_2 \leq 500$) separated by a space. The points (x_1, y_1) and (x_2, y_2) are diagonally opposite corners of a rectangle.

OUTPUT

Output the smallest number of times the pencil must be raised to draw the rectangles.

EXAMPLES

input 1 0 0 10 10 **output** 0	**input** 1 -5 -5 5 5 **output** 1	**input** 5 1 1 4 4 3 3 6 6 4 4 5 5 5 0 8 3 6 1 7 2 **output** 2

Mirko owns a bakery on the edge of the city. Because of the financial crisis, Mirko's monthly gas bill for the ovens has become too large so he decided to steal gas from a large pipeline near his bakery.

The land surrounding the bakery can be modelled by an R×C grid so that the pipeline is in the first column and Mirko's bakery in the last column.

Mirko will connect pipes to the gas pipeline and his bakery. Some parcels of land are inaccessible and Mirko cannot put pipes through them.

Every path from the pipeline to the bakery starts in a cell in the **first column** in the grid, ends in the **last column**, and each cell on the path is connected to the cell diagonally up-right, right or diagonally down-right.

In order to pull as much gas from the pipeline as possible, Mirko wants to get as many paths to his bakery as possible. The paths may not cross nor touch. In other words, **at most one path** may pass through any cell.

Write a program that calculates the largest number of paths Mirko can pull.

INPUT

The first line contains two integers R and C ($1 \leq R \leq 10000$, $5 \leq C \leq 500$), the dimensions of the grid.

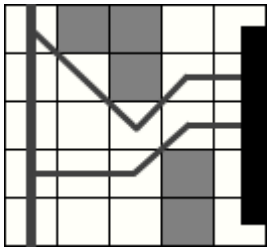
Each of the following R lines contains a string of C characters, each being a period '.' or the letter 'x'. The letter 'x' represents an inaccessible parcel of land. The first and last cells in each row will be clear.

OUTPUT

Output the largest number of paths Mirko can pull from the pipeline to the bakery.

EXAMPLES

input	input
5 5 .xx.. ..x.. ....x. ...x.	6 10 ..x....... .....x.... .x....x... ...x...xx. ....x.....
output	output
2	5



The grid from the first example, including one possible configuration of paths.