problem	utrka	labudovi	nagibni

source file	`utrka.pas`
	`utrka.c`
	`utrka.cpp`
		`labudovi.pas`
		`labudovi.c`
		`labudovi.cpp`
			`nagibni.pas`
			`nagibni.c`
			`nagibni.cpp`

input data	stdin		
output data	stdout		
time limit (Athlon 64 3000+)	1 sec	2 sec	1 sec
memory limit (heap)	32 MB	64 MB	32 MB
memory limit (stack)	8 MB	16 MB	8 MB
points	**50**	**70**	**80**
	200		

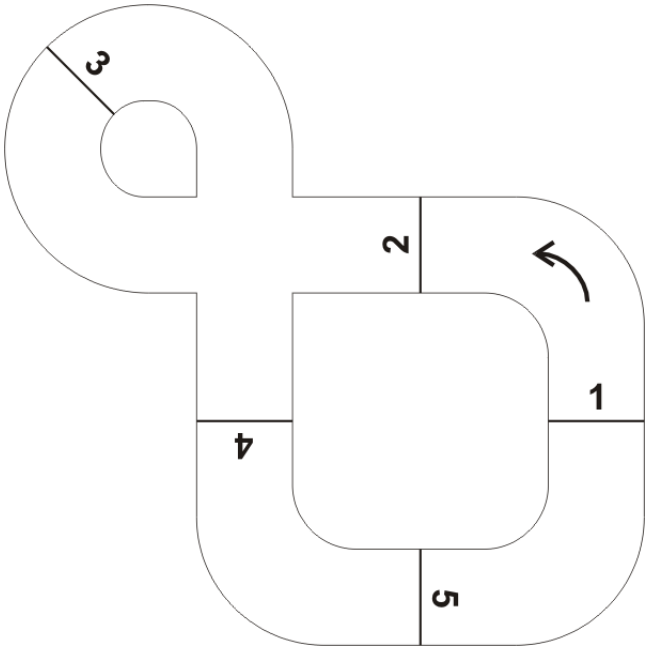
We have to write a computer program to rank the drivers during a car race.

Along the closed racetrack there are K **checkpoints** (designated with numbers from 1 to K) with judges at the each checkpoint. When some driver passes through the checkpoint, judge sends a message to the computer system with the number of the checkpoint and the number of the driver (drivers are designated with numbers from 1 to N).

Race starts just before the **first checkpoint** i.e. drivers pass through that checkpoint immediately after the start.

Write a program that will calculate the **current ranking** of the drivers, given all of the messages sent by the judges.

If two drivers have regularly passed the same number of checkpoints, the driver who passed the last point earlier is ranked higher.



The figure above depicts a racetrack with five checkpoints. Drivers must pass the points in the right order. If a driver passes other points between two consecutive checkpoints, we ignore those extra passes.

input data

First line of input contains three integers K, N and M, $1 \leq K \leq 100$, $1 \leq N \leq 100$, $1 \leq M \leq 10000$.

Each of the following M lines contains two integers X and Y. They represent a message from a judge saying that the driver with number X passed by the checkpoint Y. Events in the input are given in chronological order.

Note: there will always be a solution for the given test data.

output data

First and only line of output should contain the final ranking of drivers in the race, after all M messages from the judges have been processed.

examples

input

```
3 3 6
3 1
1 1
2 1
3 2
3 3
2 2
```

output

```
3 2 1
```

input

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

output

```
1 2
```

input

```
4 4 22
3 1
2 1
4 2
4 3
4 4
4 1
1 1
4 2
1 2
1 3
3 2
3 3
2 2
2 3
3 4
2 4
3 1
1 4
2 1
4 3
2 2
3 2
```

output

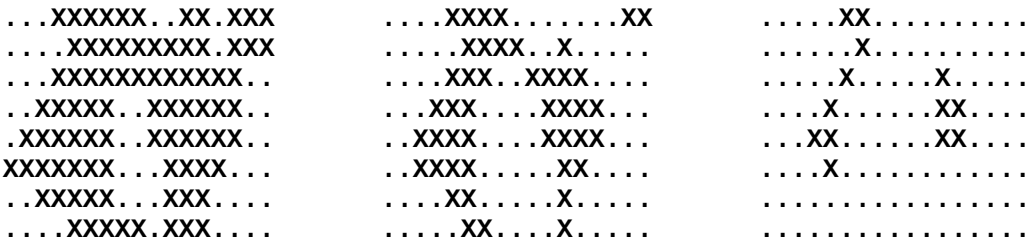
```
2 3 1 4
```

Two swans are living on a lake but they are separated with ice that covers parts of the lake.

Lake is rectangular in shape and consists of squares arranged in R rows and C columns. Some squares are covered with ice.

Lake is gradually defrosting – **in one day** all of the squares covered with ice that are **in touch with water** melt and turn into water. We consider two squares to be in touch if they are neighbors horizontally or vertically (but not diagonally).

The following figure depicts the lake from the third example:



Swans can move only on water squares in horizontal and vertical (but not diagonal) direction.

Write a program that will calculate after **how many days** the swans will be able to meet each other.

input data

First line of input contains two integers R and C, $1 \leq R, C \leq 1500$.

Each of the following R lines contains a sequence of C characters, the description of the lake at the beginning: '**.**' (dot) denotes a water square, '**X**' denotes an ice-covered square, and '**L**' denotes a square with a swan.

output data

First and only line of output should contain the number of days from the problem statement.

examples

input

```
10 2
.L
..
XX
XX
XX
XX
XX
XX
..
.L
```

output

```
3
```

input

```
4 11
..XXX...X..
.X.XXX...L.
....XXX..X.
X.L..XXX...
```

output

```
2
```

input

```
8 17
...XXXXXX..XX.XXX
....XXXXXXXXX.XXX
...XXXXXXXXXXX..
..XXXXX.LXXXXXX..
.XXXXXX..XXXXXX..
XXXXXXX...XXXX...
..XXXXX...XXX....
....XXXXX.XXXL...
```

output

```
2
```

Railway network infrastructure in one country consists of a certain numbers of individual railway segments.

Each segment connects two different towns. We know the lengths of all segments, and the network is such that between any two towns there is a **unique path** connecting them.

Recently, the management of the railway company decided to replace their old trains with ultramodern and fast trains imported from Japan.

A route of a train consists of a sequence of **two or more** towns, with all neighboring towns on the route connected by a segment. **Length of the route** is defined as the **sum of the lengths** of all the segments included. Because trains are very fast it is not allowed that two different routes pass through the same town.

Write a program that will determine the new routes in the network under the following restrictions: **Exactly one** route has to pass through each town, and **sum of the lengths** of all the routes should be **as large as possible**.

input data

First line of input contains an integer N, $1 \leq N \leq 2000$, the number of towns. Towns are designated with numbers from 1 to N.

Following N-1 lines contain the descriptions of the segments. Each line contains three integers A, B and C representing a segment between town A and town B of length C, $1 \leq C \leq 1{,}000{,}000$.

Note: there will always exist a solution for the given test data.

output data

First and only line of output should contain only the maximal sum from the problem statement.

examples

input

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

output

```
21
```

input

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

output

```
20
```

input

```
9
1 2 1
2 4 2
3 4 1
4 5 5
6 5 1
5 7 2
7 8 1
8 9 2
```

output

```
11
```