
problems

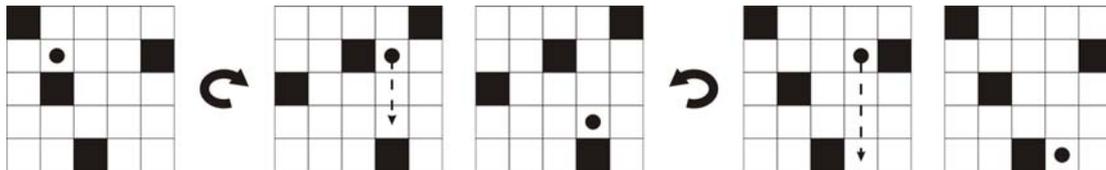
problem	lopta	nogomet	ribari
source file	lopta.pas lopta.c lopta.cpp	nogomet.pas nogomet.c nogomet.cpp	ribari.pas ribari.c ribari.cpp
input data	stdin		
output data	stdout		
time limit (Athlon 64 3000+)	1 sec		
memory limit (heap)	32 MB		
memory limit (stack)	8 MB		
points	50	70	80
	200		

lopta

Game board consists of $N \times N$ small squares arranged in N rows and N columns. A square can be empty or occupied by a wall. There is a ball in one empty square on the board. Ball is under the influence of gravity, so it **must always lie on a wall or on the floor** of the game board.

We can rotate the game board 90 degrees clockwise or counterclockwise. Walls and the ball are rotating together with the whole board. After the rotation, the ball **drops down** (influenced by gravity) on the first wall under it (or on the floor of the board).

Example of a board first rotated clockwise and then counterclockwise:



Write a program that will determine the layout of the board after a sequence of given rotations.

input data

First line of input contains integers N and K , $1 \leq N \leq 1000$, $1 \leq K \leq 500,000$.

Following N lines contain the initial description of the board: '.' (dot) denotes empty square, 'X' denotes wall, and 'L' the initial position of the ball.

In each of the following K lines there is one character, either 'L' or 'D', representing the direction of rotation in the corresponding step. 'L' means counterclockwise and 'D' means clockwise rotation.

lopta

output data

N lines of output should contain the layout of the board after all K given rotations.

examples

input

```
6 2
....XX
X.....
.....
..L...
.XXX..
.....
L
D
```

output

```
....XX
X.....
.....
.XXX..
L.....
```

input

```
10 7
.....
XXXXXXXXXX.
..X.....X.
..X.....X.
.....X.
.....X.
..X.....X.
..X.....X.
..X.....X.
.XXXXXXXX.
L.....
L
L
L
D
D
L
```

output

```
.....X.
.X.....X.
.X....XXX.
.XXX....X.
.X.....X.
.X.....X.
.X.....X.
.X...L..X.
.XXXXXXXX.
.....
```

input

```
10 8
...XXX..XX
....XX...X
L....X..XX
X..X..X...
.....X..
..X.....X.
.X.....
X..X.....
...X.....
.....X....
D
L
D
L
D
D
D
D
L
```

output

```
....X.....
.....X...
.....X...X
.....X..
.X.....X..
..X.....
...X..X..X
XX..X.....
X...XX....
XXL..XXX...
```

nogomet

We have to create a computer program that will calculate the ranking of football teams after a certain number of matches in the league have been played.

For each played match, the team gets **3 points** for a win, **1 point** for a draw and **0 points** for a loss.

The ranking is determined by the following rules:

- teams are **sorted descendingly** according to the total number of points obtained in all matches,
- if some teams (**but not all**) have the same number of points, we construct a new league consisting of these teams only. In this league we take into account only points from the matches played between these teams. Then, inside this new league we apply the same rules for determining the ranking.
- if all of the teams have equal number of points, ranking is calculating in the following way: first we compare goal difference, then the number of goals scored, then the number of wins; if some teams are still equal, team with smaller number will be ranked better. Goal difference, number of goals scored and number of wins are calculated from the **all matches** of the initial league.

Write a program that will determine the ranking of teams in the league.

input data

First line of input contains two integers N and K, $1 \leq N \leq 100$, $1 \leq K \leq 1000$, number of teams and number of played matches. Teams are designated with numbers from 1 to N.

Following K lines contain the description of played matches, in the format 'A B C:D'. This means that teams A and B have played their match, and team A scored C goals while team B scored D goals, $0 \leq C, D \leq 9$.

output data

First and only line of output should contain ranking of the league, from the best team to the worst team.

examples

input

```
3 2
1 2 0:3
2 3 3:1
```

output

```
2 3 1
```

input

```
3 3
1 2 1:3
2 1 2:2
1 3 4:0
```

output

```
2 1 3
```

input

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

output

```
2 3 1 5 4
```

ribari

In one small country, most of the citizens are fishermen, and all the towns in that country are situated on the straight line road along the straight line seaside.

Fishermen in the towns caught great amounts of fish, but they don't like the fish like they used to, so they decided to adopt poor and hungry children from the neighboring country.

One long straight road runs along the seaside connecting all towns. Hence, each town (except the first and the last) is directly connected to both neighboring towns. One child in one year **eats one tone** of fish. Amount of caught fish in some town can be eaten in that town or can be transported to other towns. During transportation there is a loss of food of **one tone for one kilometer** travelled because of taxes along the road.

We want that each town adopts the **equal** number of poor kids. Write a program that will determine the **maximal number** of kids that can be adopted in each town with the condition that all of them can be fed with the fish caught and cleverly transported.

input data

First line of input contains an integer N , $1 \leq N \leq 100,000$, the number of towns.

Each of the following N lines contains two integers A and B , $1 \leq A \leq 1,000,000,000$, $0 \leq B \leq 1,000,000,000$, they denote the position of the town and the annual production of fish respectively. Towns will be sorted ascendingly according to the position on the road.

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

output data

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

examples

input	input	input
3	3	4
1 0	5 70	20 300
2 21	15 100	40 400
4 0	1200 20	340 700
		360 600
output	output	output
6	20	415