The government is planing to build a walkway for tourists in the middle of an oak forest. The forest can be represented as plane with N special lattice points representing oaks.

The walkway is represented as a rectangle with sides parallel to the axes. If the sides of walkway rectangle intersect any oak lattice points, such oaks need to be axed down. Oaks inside the rectangle do not represent problems and need not be cut down.

Ljubo is the state secretary of Forestry and an passionate nature lover, so he ordered the secretary of Tourism to provide him with a list of P possible rectangle walkways that are attractive enough to draw in tourists.

Ljubo plans to select the walkway that needs the smallest amount of oak trees to be cut down. Since we also like trees, would you be so kind and write a program that will determine the number of oaks that will be cut down for each walkway. Remember only the oaks **intersecting the sides** of the rectangle need to be cut.

INPUT

The first line of input contains one integer N ($1 \leq N \leq 300\,000$), number of oaks.

The next N lines contain two integers each X i Y ($1 \leq X, Y \leq 10^9$) coordinates of oaks. There will be at most one oak on each lattice point.

The next line contains one integer P ($1 \leq P \leq 100\,000$), number of walkways.

The next P lines contain four integers each X1, Y1, X2 i Y2 ($1 \leq X1 < X2 \leq 10^9$, $1 \leq Y1 < Y2 \leq 10^9$) coordinates of the lower left (X1, Y1) and upper right (X2, Y2) corner of the rectangle.

OUTPUT

Output P integers, one per line, the number of oaks that need to be cut down for each walkway in the order they are presented in the input.

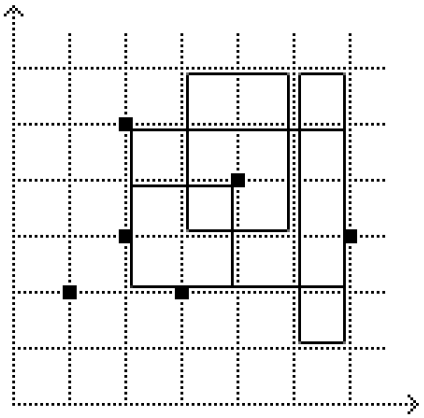
SCORING

Test data worth 30 points has ($1 \leq X1 < X2 \leq 10^3$, $1 \leq Y1 < Y2 \leq 10^3$).
Test data worth 60 points has ($1 \leq X1 < X2 \leq 10^6$, $1 \leq Y1 < Y2 \leq 10^6$).

SAMPLE TEST CASES

input

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



output

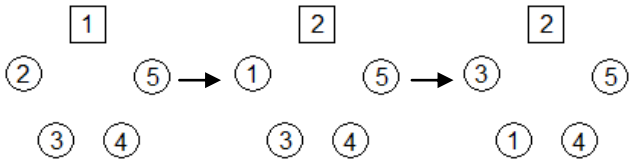
```
3
4
0
1
```

During meetings of young mathematicians a frequent pastime is the Prime Number Circle. For this task, we refer to mathematicians in the circle with numbers 1 to N.

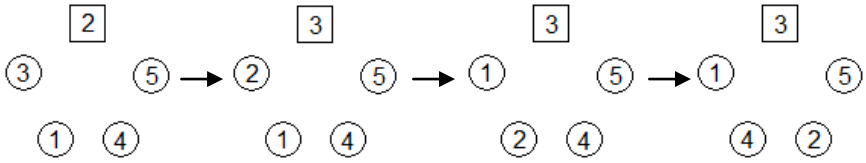
Before the game starts we first draw N-1 circles and one square on the pavement arranged in a big circle. The player numbered 1 stands in the square. All other players stand in the circles, starting with the player 2 in a counterclockwise fashion facing towards the middle of the big circle.

The game consists of K rounds. In the i-th round the person standing in the square jumps up, says "It's me!" and then swaps places with the person standing on his right side p_k times, where p_k is the k-th prime. For example for N = 5 and K = 3 the following three rounds occur:

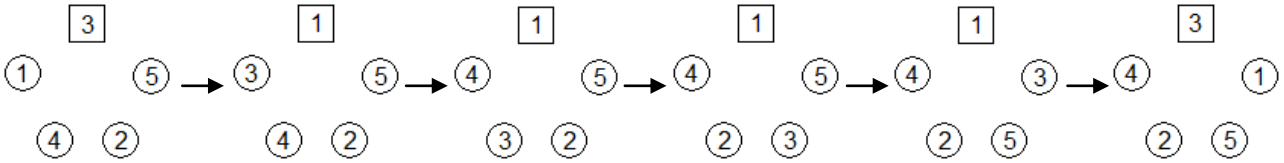
1. round:



2. round:



3. round:



Write a program that will for given N, K and A determine the neighbours of the player numbered A at the end of the game.

INPUT

The first and only line contains three integers N, K and A. ($1 \leq A \leq N$), the number of players, rounds and the selected player.

SCORING

Test data is divided into four groups each worth 25 points, with the following constraints:

First group: ($3 \leq N \leq 1000$, $1 \leq K \leq 1000$).

Secong group: ($3 \leq N \leq 1000$, $1 \leq K \leq 50000$).

Third group: ($3 \leq N \leq 50000$, $1 \leq K \leq 50000$).

Fourth group: ($3 \leq N \leq 5000000$, $1 \leq K \leq 500000$).

OUTPUT

The first and only line of output should contain two integers, the numbers on the right and left neighbours of the player numbered A at the end of the game.

SAMPLE TEST CASES

input	input	input
5 3 1	5 3 2	5 4 5
output	**output**	**output**
3 5	5 4	3 2

Scientists on Antarctica have found a new species! They extracted one sample and took it to the laboratory for testing.

They quickly noticed the species reproduces quite often and that only one parent is required for reproduction. However after the parent reproduces twice, it becomes sterile and cannot reproduce again.

In spite of this the number of specimens in the laboratory skyrocketed and the need for family trees appeared.

They decided to draw the tree in a simple text editor using following conventions:

The names of the specimens will be written inside nice boxes using characters '-', '|' and 'o'. The center point of the upper and lower border will be marked by the character '+'. If the length of the border is even, '+' will be on the left of the two center points.

```
o--+--o        o----+----o        o-+--o
|anton|        |anamarija|        |pero|
o--+--o        o----+----o        o-+--o
```

The boxes will be connected with links. One link connects two or more boxes on their respective '+' characters, with the parent specimen placed above it's children. The boxes and links must not overlap.

```
+              +                 +
|              |                 |
o           o--o--o           o-----o-----o
|           |     |           |           |
+           +     +           +           +
```

If the parent has one child, a point to point (leftmost example) link is used. If the parent has two children, a branching link is used, **with the older child on the left, and the younger on the right.**

Branching links may be expanded in the horizontal direction as long as the number of '-' characters on both sides stays equal. Links **cannot be expanded vertically.**

Don't worry, you will not be asked to draw the tree, only determine the number of characters required to draw it. **Space characters are not counted**, only '-', '|', '+' , 'o' and letters in the names.

INPUT

The first line of input contains one integer N ($1 \leq N \leq 300\,000$), number of specimens in the laboratory.

The specimens are marked with numbers 1 to N in the order of birth, with the oldest marked 1 and youngest marked N.

The next N lines contain birth notes of all specimens in the order of birth. Each specimen (except the first whose parent is unknown) is described with two pieces of information:

- *name* - sequence of at most 20 lowercase english alphabet characters
- *parent* – one integer denoting the parent of this specimen

OUTPUT

The first and only line of input should contain the minimal number of characters needed to draw the family tree.

SCORING

Test data worth 50 points has N < 30.

Test data worth 75 points has N < 3000.

SAMPLE TEST CASES

input

```
3
adam
kain 1
abel 1
```

output

```
64
```

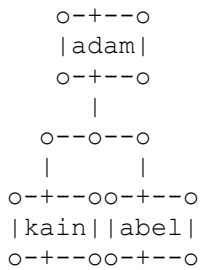
input

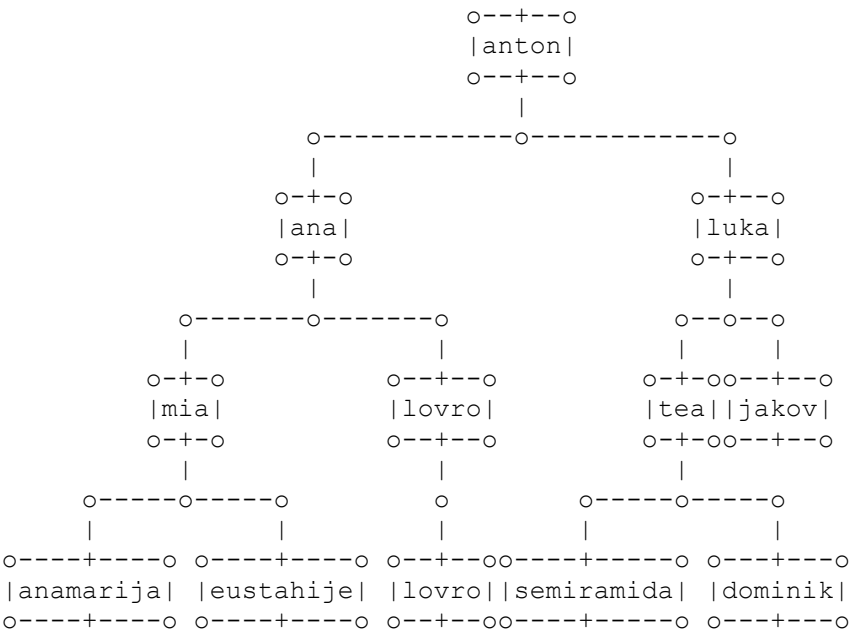
```
12
anton
ana 1
luka 1
mia 2
tea 3
jakov 3
semiramida 5
dominik 5
anamarija 4
eustahije 4
lovro 2
lovro 11
```

output

```
371
```

Sample description:





Counting the characters gives 64 and 371.

Two robots are lost in a warehouse. The robots are numbered 1 and 2. The warehouse is a grid with R rows and C columns with each field either blocked or free. The robots are controlled through radio command. Each command consists of two pieces of data:

- *robot* - 1 or 2, the number of robot we are moving
- *direction* - character 'U', 'D', 'L' or R representing the direction (up, down, left, right) we want to move the robot in.

If the destination field is blocked, taken by another robot or outside the warehouse, nothing happens and the robot stays where he is. If the field is free, the robot moves in it.

The robots are equipped with GPS devices, however due to malfunctions during deployment we cannot receive the exact location of the robots, only their M**anhattan distance** to each other. If the robots are on fields (r_1, c_1) and (r_2, c_2), then their Manhattan distance is $|r_1 - r_2| + |c_1 - c_2|$.

After each command, unsuccessful or successful, the only information we know is the current distance.

Robots are currently in the warehouse on different, unoccupied locations. Write a program that will issue a series of command required to place both robots on two special extraction points in the warehouse. The commands are issued by writing on the standard output. After each command you must read the current distance on the standard input.

The warehouse will be such that all free fields are connected.

INTERACTION

Before interacting with the robots, a few input data is given.

The first line contains integers R and C ($2 \leq R, C \leq 200$), number of rows and columns in the warehouse.

The next R rows contain C characters each. Only characters '**.**', '#' and 'x' will appear. '.' represents a free field, '#' blocked field and 'x' one of the two extraction fields. There will be exactly two 'x' characters.

The next line contains the starting Manhattan distance.

After loading all input data, you may begin issuing command to the robots using standard input. Each command must be formatted "robot direction" followed by newline character. After each command you **must** flush the standard output. Use fflush(stdout) in C/C++ or flush(StdOut) in Pascal.

When you are done issuing commands, print "0" on a line by itself and **exit your program with return code 0.**

SCORING

Test cases worth 40 points do not contain blocked fields.

Test cases worth 80 points have R and $C \leq 50$.

TESTING

You may test your solutions using the TEST interface in the evaluation system. For each test a input file must be provided. The input file must follow the following formatting:

The first line of input must contain two integers R and C ($2 \leq R, C \leq 200$).

The following R lines must contain C characters each. The characters can be only '.', '#', 'x', '1' and '2'. There must be exactly two 'x' characters. Characters '1' and '2' denote the starting locations of the robots. There must be exactly one of each present in the input. Of course, the characters '1' and '2' will be converted to '.' when presented as input to your solution.

An example of such input file is:

```
4 5
##x1.
.##..
.....
2...x
```

To test your code on the evaluation system, you must first submit your solution using the SUBMIT interface and then use the TEST interface for testing.