A. PawnChess

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Galois is one of the strongest chess players of Byteforces. He has even invented a new variant of chess, which he named «PawnChess».

This new game is played on a board consisting of 8 rows and 8 columns. At the beginning of every game some black and white pawns are placed on the board. The number of black pawns placed is not necessarily equal to the number of white pawns placed.

Lets enumerate rows and columns with integers from 1 to 8. Rows are numbered from top to bottom, while columns are numbered from left to right. Now we denote as (r, c) the cell located at the row r and at the column c.

There are always two players A and B playing the game. Player A plays with white pawns, while player B plays with black ones. The goal of player A is to put any of his pawns to the row 1, while player B tries to put any of his pawns to the row 8. As soon as any of the players completes his goal the game finishes immediately and the succeeded player is declared a winner.

Player A moves first and then they alternate turns. On his move player A must choose exactly one white pawn and move it one step upward and player B (at his turn) must choose exactly one black pawn and move it one step down. Any move is possible only if the targeted cell is empty. It's guaranteed that for any scenario of the game there will always be at least one move available for any of the players.

Moving upward means that the pawn located in (r, c) will go to the cell $(r - 1, c)$, while moving down means the pawn located in (r, c) will go to the cell $(r + 1, c)$. Again, the corresponding cell must be empty, i.e. not occupied by any other pawn of any color.

Given the initial disposition of the board, determine who wins the game if both players play optimally. Note that there will always be a winner due to the restriction that for any game scenario both players will have some moves available.

Input

The input consists of the board description given in eight lines, each line contains eight characters. Character '`B`' is used to denote a black pawn, and character '`W`' represents a white pawn. Empty cell is marked with '`.`'.

It's guaranteed that there will not be white pawns on the first row neither black pawns on the last row.

Output

Print '`A`' if player A wins the game on the given board, and '`B`' if player B will claim the victory. Again, it's guaranteed that there will always be a winner on the given board.

Sample test(s)

input

```
........
........
.B....B.
....W...
........
..W.....
........
........
```

output

```
A
```

input

```
..B.....
..W.....
......B.
........
.....W..
......B.
........
........
```

output

```
B
```

Note

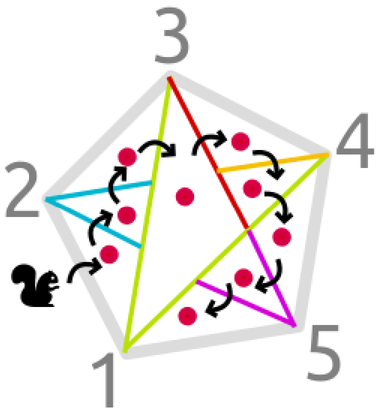
In the first sample player A is able to complete his goal in 3 steps by always moving a pawn initially located at $(4, 5)$. Player B needs at least 5 steps for any of his pawns to reach the row 8. Hence, player A will be the winner.

B. The Monster and the Squirrel

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Ari the monster always wakes up very early with the first ray of the sun and the first thing she does is feeding her squirrel.

Ari draws a regular convex polygon on the floor and numbers it's vertices $1, 2, ..., n$ in clockwise order. Then starting from the vertex 1 she draws a ray in the direction of each other vertex. The ray stops when it reaches a vertex or intersects with another ray drawn before. Ari repeats this process for vertex $2, 3, ..., n$ (in this particular order). And then she puts a walnut in each region inside the polygon.



Ada the squirrel wants to collect all the walnuts, but she is not allowed to step on the lines drawn by Ari. That means Ada have to perform a small jump if she wants to go from one region to another. Ada can jump from one region P to another region Q if and only if P and Q **share a side or a corner**.

Assuming that Ada starts from outside of the picture, what is the minimum number of jumps she has to perform in order to collect all the walnuts?

Input

The first and only line of the input contains a single integer n $(3 \le n \le 54321)$ - the number of vertices of the regular polygon drawn by Ari.

Output

Print the minimum number of jumps Ada should make to collect all the walnuts. Note, that she **doesn't need** to leave the polygon after.

Sample test(s)

input
```
5
```
output
```
9
```

input
```
3
```
output
```
1
```

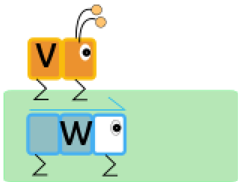
Note

One of the possible solutions for the first sample is shown on the picture above.

C. The Big Race

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Vector Willman and Array Bolt are the two most famous athletes of Byteforces. They are going to compete in a race with a distance of L meters today.



Willman and Bolt have exactly the same speed, so when they compete the result is always a tie. That is a problem for the organizers because they want a winner.

While watching previous races the organizers have noticed that Willman can perform **only** steps of length equal to w meters, and Bolt can perform **only** steps of length equal to b meters. Organizers decided to slightly change the rules of the race. Now, at the end of the racetrack there will be an abyss, and the winner will be declared the athlete, who manages to run farther from the starting point of the the racetrack (which is not the subject to change by any of the athletes).

Note that none of the athletes can run infinitely far, as they both will at some moment of time face the point, such that only one step further will cause them to fall in the abyss. In other words, the athlete **will not** fall into the abyss if the total length of all his steps will be less or equal to the chosen distance L.

Since the organizers are very fair, the are going to set the length of the racetrack as an integer chosen randomly and uniformly in range from 1 to t (both are included). What is the probability that Willman and Bolt tie again today?

Input

The first line of the input contains three integers t, w and b ($1 \le t, w, b \le 5 \cdot 10^{18}$) — the maximum possible length of the racetrack, the length of Willman's steps and the length of Bolt's steps respectively.

Output

Print the answer to the problem as an irreducible fraction $\frac{p}{q}$. Follow the format of the samples output.

The fraction $\frac{p}{q}$ (p and q are integers, and both $p \ge 0$ and $q > 0$ holds) is called irreducible, if there is no such integer $d > 1$, that both p and q are divisible by d.

Sample test(s)

input
```
10 3 2
```
output
```
3/10
```

input
```
7 1 2
```
output
```
3/7
```

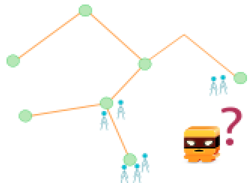
Note

In the first sample Willman and Bolt will tie in case 1, 6 or 7 are chosen as the length of the racetrack.

D. Super M

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Ari the monster is not an ordinary monster. She is the hidden identity of Super M, the Byteforces' superhero. Byteforces is a country that consists of n cities, connected by $n - 1$ bidirectional roads. Every road connects exactly two distinct cities, and the whole road system is designed in a way that one is able to go from any city to any other city using only the given roads. There are m cities being attacked by humans. So Ari... we meant Super M have to immediately go to each of the cities being attacked to scare those bad humans. Super M can pass from one city to another only using the given roads. Moreover, passing through one road takes her exactly one kron - the time unit used in Byteforces.



However, Super M is not on Byteforces now - she is attending a training camp located in a nearby country Codeforces. Fortunately, there is a special device in Codeforces that allows her to instantly teleport from Codeforces to any city of Byteforces. The way back is too long, so for the purpose of this problem teleportation is used exactly once.

You are to help Super M, by calculating the city in which she should teleport at the beginning in order to end her job in the minimum time (measured in krons). Also, provide her with this time so she can plan her way back to Codeforces.

Input

The first line of the input contains two integers n and m $(1 \le m \le n \le 123456)$ - the number of cities in Byteforces, and the number of cities being attacked respectively.

Then follow $n - 1$ lines, describing the road system. Each line contains two city numbers u_i and v_i $(1 \le u_i, v_i \le n)$ - the ends of the road i.

The last line contains m distinct integers - numbers of cities being attacked. These numbers are given in no particular order.

Output

First print the number of the city Super M should teleport to. If there are many possible optimal answers, print the one with the lowest city number.

Then print the minimum possible time needed to scare all humans in cities being attacked, measured in Krons.

Note that the correct answer is always unique.

Sample test(s)

input

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

output

```
2
3
```

input

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

output

```
2
4
```

Note

In the first sample, there are two possibilities to finish the Super M's job in 3 krons. They are:

$2 \to 1 \to 3 \to 7$ and $7 \to 3 \to 1 \to 2$.

However, you should choose the first one as it starts in the city with the lower number.

E. BCPC

time limit per test: 4 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

BCPC stands for Byteforces Collegiate Programming Contest, and is the most famous competition in Byteforces.

BCPC is a team competition. Each team is composed by a coach and three contestants. Blenda is the coach of the Bit State University(BSU), and she is very strict selecting the members of her team.

In BSU there are n students numbered from 1 to n. Since all BSU students are infinitely smart, the only important parameters for Blenda are their reading and writing speed. After a careful measuring, Blenda have found that the i-th student have a **reading** speed equal to r_i (words per minute), and a **writing** speed of w_i (symbols per minute). Since BSU students are very smart, the measured speeds are sometimes very big and Blenda have decided to subtract some constant value c from all the values of reading speed and some value d from all the values of writing speed. Therefore she considers $r_i' = r_i - c$ and $w_i' = w_i - d$.

The student i is said to *overwhelm* the student j if and only if $r_i' \cdot w_j' > r_j' \cdot w_i'$. Blenda doesn't like fights in teams, so she thinks that a team consisting of three distinct students i, j and k is *good* if i overwhelms j, j overwhelms k, and k overwhelms i. Yes, the relation of overwhelming is not transitive as it often happens in real life.

Since Blenda is busy preparing a training camp in Codeforces, you are given a task to calculate the number of different good teams in BSU. Two teams are considered to be different if there is at least one student that is present in one team but is not present in the other. In other words, two teams are different if the sets of students that form these teams are different.

Input

In the first line of the input three integers n, c and d ($3 \le n \le 345678$, $1 \le c, d \le 10^9$) are written. They denote the number of students Blenda can use to form teams, the value subtracted from all reading speeds and the value subtracted from all writing speeds respectively.

Each of the next n lines contains two integers r_i and w_i ($0 < r_i, w_i \le 10^9$, $|r_i - c| + |w_i - d| > 0$). There are no two students, such that both their reading and writing speeds coincide, i.e. for every $i \ne j$ condition $|r_i - r_j| + |w_i - w_j| > 0$ holds.

Output

Print the number of different teams in BSU, that are good according to Blenda's definition.

Sample test(s)

input

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

output

```
4
```

input

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

output

```
11
```

Note

In the first sample the following teams are good: $(i = 1, j = 2, k = 3)$, $(i = 2, j = 5, k = 1)$, $(i = 1, j = 4, k = 3)$, $(i = 5, j = 1, k = 4)$.

Note, that for example the team $(i = 3, j = 1, k = 2)$ is also good, but is considered to be the same as the team $(i = 1, j = 2, k = 3)$.