ICPC SOUTH PACIFIC DIVISIONALS LEVEL A

OCTOBER 15, 2022

Contest Problems

A: A Powerful Tower
B: Bubble Bath Sort
C: City Planning
D: Dee's NFTs
E: Efficient City
F: Floating Icebergs
G: GPU Meme
H: Hopeless Bingo
I : Iguana Wedding
J : Jumpy Children
K: Kiwis, Snakes, and Ladders
L: Locating Ice Cream

This contest contains twelve problems. Good luck.

For problems that state "*Your answer should have an absolute or relative error of less than* 10^{-9}", your answer, x, will be compared to the correct answer, y. If $|x - y| < 10^{-9}$ or $\frac{|x-y|}{|y|} < 10^{-9}$, then your answer will be considered correct.

Definition 1

For problems that ask for a result modulo m:
If the correct answer to the problem is the integer b, then you should display the unique value a such that:

- $0 \leq a < m$
  and
- $(a - b)$ is a multiple of m.

Definition 2

A string $s_1 s_2 \cdots s_n$ is lexicographically smaller than $t_1 t_2 \cdots t_\ell$ if

- there exists $k \leq \min(n, \ell)$ such that $s_i = t_i$ for all $1 \leq i < k$ and $s_k < t_k$
  or
- $s_i = t_i$ for all $1 \leq i \leq \min(n, \ell)$ and $n < \ell$.

Definition 3

- Uppercase letters are the uppercase English letters ($A, B, \ldots, Z$).
- Lowercase letters are the lowercase English letters ($a, b, \ldots, z$).

Definition 4

Unless otherwise specified, the distance between two points (x_0, y_0) and (x_1, y_1) is defined as its Euclidean distance:

$$\sqrt{(x_0 - x_1)^2 + (y_0 - y_1)^2}.$$

Problem A

A Powerful Tower

Time limit: 2 seconds

Marcell has built a grid of blocks with R rows and C columns. Every block has a lowercase letter written on it.

A contiguous subrectangle of the grid is called a *tower* if it includes all rows of the grid and the letters read left-to-right in each row are the same as the other rows. The *power* of a tower is the number of columns it spans.

What is the maximum power of all towers in the grid?

Input

The first line of input contains two integers, R ($1 \leq R \leq 2\,000$), which is the number of rows, and C ($1 \leq C \leq 2\,000$), which is the number of columns.

The next R lines of input each contain C lowercase letters. The jth letter on the ith line is letter at location (i, j) in the grid of blocks.

Output

If there are no towers, display 0. Otherwise, display the maximum power of all towers in the grid.

Sample Input 1	Sample Output 1
2 3 abc xbc	2

Sample Input 2	Sample Output 2
2 3 abc xcb	0

Sample Input 3	Sample Output 3
1 4 xyfh	4

Sample Input 4	Sample Output 4
3 3 abc efg hij	0

This page is intentionally left (almost) blank.

Problem B
Bubble Bath Sort
Time limit: 10 seconds

Bubble Baths & Beyond has installed a robot to sort their products. All their bubble baths have a serial number by which they are to be sorted, and are in a line from position 1 to position n.

In each round, the robot goes down the line from position 1 first and swaps each pair of adjacent bubble baths if they are out of order. It first compares the baths at positions 1 and 2, then 2 and 3, then 3 and 4, and so on. After comparing positions $n-1$ and n, the robot will stop if the serial numbers are in sorted order. Otherwise, it will return to position 1 and start the next round.

Given the list of the n distinct serial numbers, Bubble Baths & Beyond wants to know how many rounds the robot will take before stopping and how many swaps it will make each round.

Input

The first line of input contains a single integer n ($1 \leq n \leq 500\,000$), which is the number of bubble baths.

The next n lines describe the serial numbers in each position ordered from 1 to n. Each of these lines contains a single integer s ($1 \leq s \leq 10^9$), which is the serial number of this bubble bath. The n serial numbers are distinct.

Output

First, display R, the number of rounds the robot spends before stopping. Then, display R more integers: the number of swaps performed in each round, in order.

Sample Input 1	Sample Output 1
3 668 549 312	2 2 1

Sample Input 2	Sample Output 2
4 711 299 367 113	3 3 1 1

Sample Input 3	Sample Output 3
2 111 222	0

This page is intentionally left (almost) blank.

Problem C
City Planning
Time limit: 5 seconds

Amanda lives in a city with lazy city planners. The city comprises N intersections with roads between them. The roads have a direction; that is, they are one-way. It is possible for a pair of intersections to have roads connecting them in both directions. It is also possible for an intersection to have a road to itself.

The lazy city planners keep a collection of M lists of intersections. The lists are labelled from 1 to M. The roads out of an intersection i are defined by a number ℓ_i, which means that there is an outgoing road from i to each intersection in list number ℓ_i from the collection of M lists.

Amanda lives at intersection 1 and works at intersection N. What is the minimum number of roads she needs to travel down to get from home to work?

Input

The first line of input contains two integers, N ($2 \leq N \leq 500\,000$), which is the number of intersections, and M ($1 \leq M \leq 500\,000$), which is the number of lists.

The next M lines describe the lists. The kth of these lines contains an integer c ($1 \leq c \leq N$) followed by c distinct integers $a_1, a_2, \ldots, a_c$ ($1 \leq a_i \leq N$), which are the intersections in list k. The total length of all lists is at most $500\,000$.

The final line of input contains N integers $\ell_1, \ell_2, \ldots, \ell_N$ ($1 \leq \ell_i \leq M$), which denote that intersection i's outgoing roadways are the list labelled ℓ_i.

Output

Display the minimum number of roads Amanda must travel. If it is impossible for her to reach work, display -1 instead.

Sample Input 1

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

Sample Output 1

```
2
```

Sample Input 2

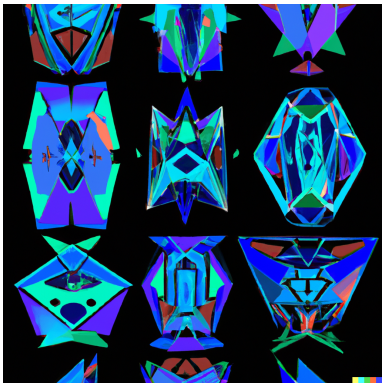
```
3 1
1 2
1 1 1
```

Sample Output 2

```
-1
```

This page is intentionally left (almost) blank.

Problem D
Dee's NFTs
Time limit: 8 seconds



NFTs have been all the rage lately! Dee wants to capitalise on the excitement and is creating a new NFT to give away to anyone that wants it. Her NFT consists of N components. Component i is generated by choosing K_i different items from a set of M_i options for that component. All of the chosen items are then combined together to form the final NFT.

Dee wants every NFT generated from her system to be unique. Two NFTs are different if any of the components have a different set of items. Dee is very optimistic and thinks that up to 10^9 people might eventually want NFTs. She is worried that there might not be enough unique NFTs that can be generated using her chosen number of components and component item counts.

The number of unique NFTs is big enough if there are more than 10^9 of them (one for everyone plus one for Dee). Help Dee determine if the number of NFTs is big enough, and find the number of unique NFTs that can be generated if it is not.

Input

The first line of input contains a single integer N ($1 \leq N \leq 10$), which is the number of NFT components.

The next N lines describe the NFT components. Each of these lines contains two integers M_i ($1 \leq M_i \leq 10^6$) and K_i ($1 \leq K_i \leq M_i$), which indicate that component i has M_i options to choose from and K_i options will be chosen.

Output

If the number of unique NFTs that could be generated is greater than 10^9, display `BIG ENOUGH`. Otherwise, display the number of unique NFTs that could be generated.

Sample Input 1

```
2
3 1
4 2
```

Sample Output 1

```
18
```

Sample Input 2

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

Sample Output 2

```
2
```

Sample Input 3

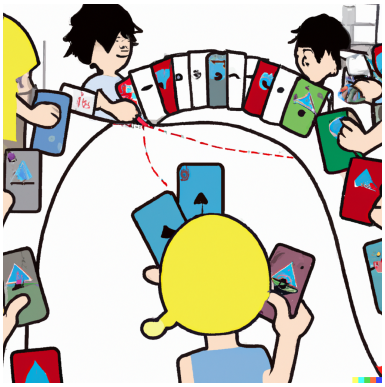
```
3
10 5
100 55
1000 555
```

Sample Output 3

```
BIG ENOUGH
```

This page is intentionally left (almost) blank.

Problem E
Efficient City
Time limit: 5 seconds



The beautiful (but boring) city of Porth has N citizens that live at distinct points on a single, straight street. The houses are numbered 1 to N. The citizens of Porth are obsessed with trading Poké-man cards.

To trade with one another, two citizens must meet at some house that can approve their trade. Every citizen c has a sphere of influence $[L_c, R_c]$ and can approve any trade where both citizens live between house numbers L_c and R_c, inclusive. Every citizen's house is part of their sphere of influence (so they can approve a trade inside their sphere of influence that they are part of).

A trade between citizen i and j is *efficient* if it is approved at some house between i and j, inclusive. The city is *highly-efficient* if every pair of citizens can find a place where they can make an efficient trade.

Given the spheres of influence, determine if Porth is highly-efficient, and if not, find a pair of citizens that cannot perform an efficient trade.

Input

The first line of input consists of a single integer N ($2 \leq N \leq 200\,000$), which is the number of citizens in Porth.

The next N lines describe their spheres of influence. The i-th of these lines contains two integers L_i and R_i ($1 \leq L_i \leq i \leq R_i \leq N$), which is the sphere of influence for the citizen at house i.

Output

If Porth is highly-efficient, display `Yes`. Otherwise, display `No` followed by two integers i and j such that citizens i and j cannot perform an efficient trade at any house.

Sample Input 1	Sample Output 1
3 1 2 2 3 1 3	Yes

Sample Input 2	Sample Output 2
4 1 2 2 3 3 3 4 4	No 1 4

Sample Input 3	Sample Output 3
3 1 1 2 2 3 3	No 1 3

This page is intentionally left (almost) blank.

Problem F
Floating Icebergs
Time limit: 1 second

Mako is trying to predict how icebergs will sink in water. She is interested in the depth at which exactly half of the iceberg will be submerged in water, so she decides to model the situation in 2D.

She represents an iceberg as a polygon without holes. The icebergs are very special, so the polygons are always convex — all the internal angles are less than 180 degrees.

The iceberg starts just above the water, so that the distance from the lowest point on the iceberg to the water level is 0 (the water level does not necessarily start at $y = 0$). Mako continuously adds water such that the water level rises by 1 on the y-axis per hour. She holds the iceberg in place, so it cannot move — there are no buoyancy effects.

Mako wants to know after how long exactly half of the iceberg's area will be submerged. Part of the iceberg is considered submerged if it is at or below the water level.

Input

The first line of the input contains a single integer N ($3 \leq N \leq 1\,000$), which is the number of points in the polygon.

The next N lines contain the points of the polygon, in order. Each of these lines contains two integers, X, Y ($-10\,000 \leq X, Y \leq 10\,000$), denoting the $x-$ and $y-$coordinate of the point. No three points will be collinear and the polygon will not have zero-length edges. The polygon edges may be given in clockwise or counter-clockwise order.

Output

Display the number of hours after which the water will submerge half of the iceberg by area. Your answer should have an absolute or relative error of less than 10^{-4}.

Sample Input 1

```
3
0 0
0 1
1 1
```

Sample Output 1

```
0.707106781187
```

Sample Input 2

```
4
-2 -2
2 -2
2 1
-2 1
```

Sample Output 2

```
1.500000000000
```

Sample Input 3

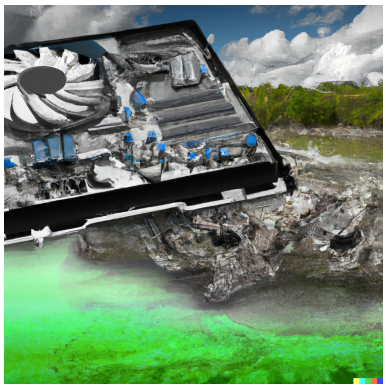
```
4
-1 -1
1 0
1 5
0 5
```

Sample Output 3

```
2.834848610088
```

This page is intentionally left (almost) blank.

Problem G
GPU Meme
Time limit: 1 second



Varun is very interested in GPUs, which he uses to convert electricity into heat and damage the environment. There are N GPUs on the market, which Varun would like to buy a subset of. Note that these are N individual GPUs and not N types of GPU, so Varun can only buy each at most once.

Each GPU has a monetary cost and an environmental cost. The total environmental costs of all GPUs that Varun buys must not exceed the legal limit. Varun will maximize the total environmental cost of the GPUs he buys, while not exceeding that legal limit. Varun has a lot of money, so can buy all the GPUs if he wants to. However, he is also frugal, so if there are multiple ways to maximize the environmental cost, Varun will choose whichever of those options minimizes the monetary cost.

What is the monetary cost of the subset of GPUs that Varun chooses?

Input

The first line of the input contains two integers N ($1 \leq N \leq 2\,000$), which is the number of GPUs, and E ($1 \leq E \leq 2\,000$), which is the legal limit of environmental cost.

The next N lines describe the GPUs. Each of these lines contains two integers M ($1 \leq M \leq 2\,000$), which is the monetary cost of this GPU, and C ($1 \leq C \leq 2\,000$), which is the environmental cost of this GPU.

Output

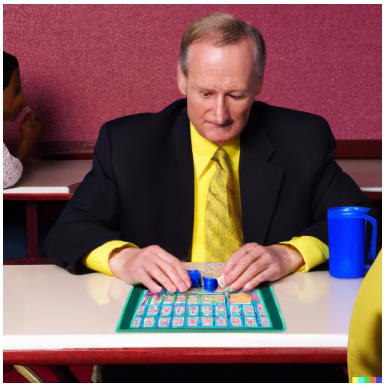
Display the monetary cost of a subset of GPUs that Varun chooses.

Sample Input 1	Sample Output 1
4 10 1 11 1 9 2 1 3 1	3

Sample Input 2	Sample Output 2
3 6 2 3 3 3 1 4	5

This page is intentionally left (almost) blank.

Problem H
Hopeless Bingo
Time limit: 1 second



Bingo is a game of chance for multiple players. Each player gets a card with a (fixed size) subset of the values from 1 to 90. A caller then draws value tokens (also from 1 to 90), calling each out when drawn (traditionally with a nickname, e.g., "66, clickety click"). Each player marks their card's values off when called, and the first player to mark all of their values off is the winner. If multiple players finish marking all their values off at the same time, they are all winners.

Many decades ago, one of your older relatives had a Bingo set but lost some of the tokens used for the random drawing, so any card with those values can never win. When someone commented that this seemed unfair, this was countered with the point that it was a game of chance that included the distribution of the cards, so the game as a whole was fair!

In tonight's game, the remaining tokens will be drawn in a particular order. Given this order, each card's status is either *winning*, *losing*, or *hopeless*. A card is winning if it is (one of) the first of the cards to have all its values drawn. A card is hopeless if it has some value that is lost (that is, not included in the calls at all). A card is losing if it is neither winning nor hopeless.

Given the order of the calls and the contents of all cards, determine the status of each card.

Input

The first line of input contains three integers N ($1 \leq N \leq 89$), which is the number of tokens left in the Bingo set, P ($2 \leq P \leq 10$), which is the number of players, and C ($1 \leq C \leq \min(30, N)$), which is the number of values on each card.

The second line contains the N distinct integers $c_1, c_2, \ldots, c_N$ ($1 \leq c_i \leq 90$), which are the tokens in the order in which they will be drawn (possibly including tokens after a card has won).

The remaining P lines describe the cards. The ith of these lines contains C distinct integers, each in the inclusive range between 1 and 90, which are the values on the ith player's card.

Output

Display the status of each player's card in the order in which they occur in the input.

Sample Input 1

```
5 6 2
2 1 3 90 66
1 3
3 2
4 66
3 90
3 90
90 3
```

Sample Output 1

```
winning
winning
hopeless
losing
losing
losing
```

Sample Input 2

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

Sample Output 2

```
hopeless
hopeless
```

This page is intentionally left (almost) blank.

Problem I
Iguana Wedding
Time limit: 1 second

The famous iguanas, Iggy and Izzy, are getting married and planning an extravagant meal for their wedding reception. Their wedding planner is putting together a meal with N courses. For each course, they are deciding between two food items to serve. All food items across all courses are unique.

Iggy and Izzy both have very particular tastes are have food restrictions for the wedding planner. They have provided their wedding planner with a list of M pairs of foods that they don't think work well together. This means that either none or one of them can be served at the wedding, but not both (regardless of which courses they are a part of). Each pair of foods will not be both items of a single course.

Can you help the wedding planner choose which food item to serve for each course such that all of Iggy and Izzy's restrictions are followed?

Input

The first line of input contains two integers N ($2 \leq N \leq 100$), which is the number of courses, and M ($1 \leq M \leq 500$), which is the number of restrictions.

The next N lines describe the courses. Each of these lines contains two nonempty strings A_i and B_i each consisting of up to 10 uppercase letters. This represents that the wedding planner must choose between food items A_i and B_i for course i. The $2N$ food items are distinct.

The next M lines describe Iggy and Izzy's restrictions. Each of these lines contains two strings C_i and D_i. These strings are different food items from the N courses and they are guaranteed to not be choices for the same course.

Output

If it is not possible for the wedding planner to choose food items to serve for each course that follows all of Izzy and Iggy's restrictions, display `No`. Otherwise, display `Yes` followed by the N food items chosen for each course (in the same order as given in the input). If there are multiple valid answers, any one will be accepted.

Sample Input 1	Sample Output 1
3 3 SALAD BREAD CHICKEN STEAK CAKE PIE SALAD STEAK CAKE BREAD STEAK PIE	Yes SALAD CHICKEN PIE

Sample Input 2	Sample Output 2
3 4 SALAD BREAD CHICKEN STEAK CAKE PIE SALAD STEAK BREAD STEAK CHICKEN CAKE PIE CHICKEN	No

Problem J
Jumpy Children
Time limit: 1 second

In the classic children's game *What time is it, Mr. Wolf?*, several children line up and start walking towards the wolf constantly asking, "what time is it, Mr. Wolf?" When ready, the wolf yells *LUNCH TIME!* and starts chasing the children back to their home base. If the wolf gets to a child before they reach home base, the child is caught. If the child gets to home base before the wolf or they arrive at home base at the same time, the child is not caught. All the children are on a single straight line between the wolf and the home base.

You are the wolf this time. When you yell *LUNCH TIME!* you turn around and see everyone's location. You know how fast you run, as well as the speeds of all of the children, but you do not remember which child runs at each of those speeds.

For example, say there were three children (Alice, Bob, and Charlie) that are 25 m, 35 m, and 40 m from home base, respectively. You (the wolf) are 100 m away from home base and run at 10 m/s. The speeds of the children are 2 m/s, 3 m/s, and 4 m/s, in some order. If Alice goes 2 m/s, Bob goes 3 m/s, and Charlie goes 4 m/s, the wolf will catch Alice and Bob before they get to home base. However, if Alice goes 3 m/s, Bob goes 2 m/s, and Charlie goes 4 m/s, then the wolf will only catch Bob. In this example, no matter how the speeds are assigned, the wolf will always be able to capture at least one child. Note that children may pass each other.

Given the children's locations and the list of speeds, what is the minimum number of children the wolf is guaranteed to capture over all assignments of speeds to children?

Input

The first line of input contains three integers N ($1 \leq N \leq 1\,000$), which is the number of children, D ($2 \leq D \leq 1\,000$), which is the distance that the wolf is from home base, and W ($1 \leq W \leq 1\,000$), which is the speed of the wolf (in m/s).

The next line describes the children's locations. This line contains N integers $\ell_1, \ell_2, \ldots, \ell_N$ ($1 \leq \ell_i < D$), which are the distances the children are from home base (in m).

The next line describes the children's speeds. This line contains N integers $s_1, s_2, \ldots, s_N$ ($1 \leq s_j \leq 1\,000$), which are the speeds of the children (in m/s), in an unspecified order.

Output

Display the minimum number of children the wolf is guaranteed to capture.

Sample Input 1	Sample Output 1
3 100 10 25 35 40 2 3 4	1

Sample Input 2	Sample Output 2
3 100 4 35 40 25 2 3 4	0

Sample Input 3	Sample Output 3
2 2 100 1 1 100 1	1

Sample Input 4	Sample Output 4
3 2 100 1 1 1 5 10 15	3

Problem K
Kiwis, Snakes, and Ladders
Time limit: 1 second

Snakes and Ladders is Janet's favourite classic board game. There are 100 cells on the 10×10 board, which are numbered 1 to 100. The cells are in the following configuration:

```
100  99  98  97  96  95  94  93  92  91
 81  82  83  84  85  86  87  88  89  90
 80  79  78  77  76  75  74  73  72  71
 61  62  63  64  65  66  67  68  69  70
 60  59  58  57  56  55  54  53  52  51
 41  42  43  44  45  46  47  48  49  50
 40  39  38  37  36  35  34  33  32  31
 21  22  23  24  25  26  27  28  29  30
 20  19  18  17  16  15  14  13  12  11
  1   2   3   4   5   6   7   8   9  10
```

Janet has played the game so many times that she has introduced a variation: Kiwis, Snakes, and Ladders.

There are four players playing the game. All four players start on cell 1. The players play in turns (player 1, then 2, then 3, then 4, then 1, then 2, and so on). A game uses a deck containing N cards, each with a number between 1 and 6 inclusive written on it. On a player's turn, they draw a card and move forward that many cells. If that card makes them go past cell 100, then they are finished and no longer take turns in the future. Otherwise, the player looks at the cell they landed on. Each cell contains one of a snake, a ladder, a kiwi, or nothing. If there is nothing in the cell they landed on, their turn is over. If there is a snake or ladder, they are teleported to the cell corresponding to that snake or ladder and then their turn is over (the contents of the new cell they teleported to are ignored). If there is a kiwi, they remain on that cell but miss their next turn (they do not draw a card nor move on a missed turn).

The game is over after a total of N turns or after all players have finished (whichever comes first). Given the board and the cards in the deck, where are each of the four players when the game is over?

Input

The first 10 lines of input describe the 10×10 board. Each of these lines contains 10 integers describing the 10 cells in this row of the board. Each cell is represented by an integer b ($-1 \leq b \leq 100$). If $b = -1$, then the cell is a kiwi, if $b = 0$, then the cell is empty. Otherwise, if $1 \leq b \leq 100$, then this is a snake/ladder that teleports you to cell number b.

The next line contains a single integer N ($1 \leq N \leq 100$), which is the number of cards in the deck.

The next line contains N integers $c_1, c_2, \ldots, c_N$ ($1 \leq c_i \leq 6$), which are the cards in the deck, in order.

Output

Display the location of each player at the end of the game, in order. If a player has finished the game, their location is 101.

Sample Input 1

```
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
9
6 6 6 6 6 6 6 6 6
```

Sample Output 1

```
19 13 13 13
```

Sample Input 2

```
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 50 0 0 0 0 0 0 0 0
0 100 0 0 0 0 0 0 0 0
9
1 6 6 6 6 6 6 6 6
```

Sample Output 2

```
101 50 13 13
```

Sample Input 3

```
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 2 3 4 5 6 7 8 9 10
1
1
```

Sample Output 3

```
2 1 1 1
```

Sample Input 4

```
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 100 0 0 0 0 0 0 0 0
9
1 1 1 1 2 2 2 2 3
```

Sample Output 4

```
101 101 101 101
```

Sample Input 5

```
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 -1 0 0 0 0 0 0 0 0
11
1 2 2 2 3 3 3 4 4 4 4
```

Sample Output 5

```
6 10 10 10
```

This page is intentionally left (almost) blank.

Problem L
Locating Ice Cream
Time limit: 7 seconds

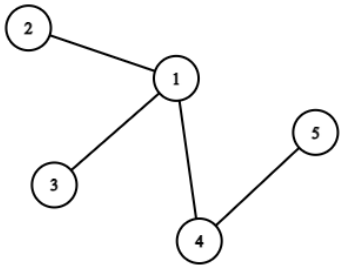
Bridget and Claude are on a road trip. They have decided on a set of cities that they may visit, but they have not decided which of them they will actually visit. The cities are connected by bidirectional roads in such a way that there is exactly one way from every city to every other city using each road at most once.

The road trip will start at some city. Every day, one of Bridget or Claude will choose one of the roads connected to the current city that has not been used before (in either direction) and then both will use that road to go to a new city. Bridget and Claude will alternate who chooses which road to take. Bridget will choose on the first, third, fifth, etc. days, while Claude will choose on the other days.

Eventually, they will arrive at a city where there are no roads to choose from. This means the road trip has ended, which Bridget and Claude celebrate with ice cream. When that happens, the person who was supposed to decide which road to take will instead choose which flavour of ice cream to buy (all flavours are available in every city).

Bridget loves ice cream. Bridget *likes* a city if the road trip can start at that city and she can guarantee that she will decide on the flavour of ice cream no matter what Claude does.

For example, consider the image. If they start at city 2, then Bridget must take the road to city 1. At this point, Claude may either take the road to city 3 or city 4. If he chooses city 4, then Bridget will be forced to go to city 5, where Claude will get to choose the flavour of ice cream. Thus, it is not guaranteed that she gets to select the flavour, so Bridget does not like city 2. However, Bridget likes city 4. If they start at city 4, then Bridget can take the road to city 1. No matter which road Claude takes, Bridget gets to choose the flavour of ice cream on the next day.



Given the layout of the roads, how many cities does Bridget like?

Input

The first line of input contains a single integer N ($1 \leq N \leq 200\,000$), which is the number of cities.

The next $N - 1$ lines describe the roads. Each of these lines contains two distinct integers x ($1 \leq x \leq N$) and y ($1 \leq y \leq N$) denoting that there is a road between cities x and y.

Output

Display the number of cities that Bridget likes.

Sample Input 1	Sample Output 1
5 1 2 1 3 1 4 4 5	2

Sample Input 2	Sample Output 2
4 1 2 2 3 2 4	3

Sample Input 3	Sample Output 3
1	1

Sample Input 4	Sample Output 4
2 2 1	0