Croatian Open Competition in Informatics

Round 1, October 17th 2020

Tasks

Task	Time limit	Memory limit	Points
Patkice	1 second	512 MiB	50
Bajka	1 second	512 MiB	70
3D Histogram	2.5 seconds	512 MiB	110
Papričice	1 second	512 MiB	110
Tenis	1 second	512 MiB	110
Total			450

Task Patkice

Not so long ago, in a distant tropical land, there lived three rubber ducks. One hot summer day while they were lying on the beach, the ducks decided to travel to a nearby island. Since the ducks like adventures, they decided to travel carried by ocean currents in an old black umbrella.

Since the ducks are experienced ocean explorers, before the voyage they will check out a map of ocean currents. On the map, the island where the ducks live is marked by a letter `'o'`. The ducks can start their voyage in any of the four directions: north – `N`, east – `E`, west – `W` and south – `S`.

Ocean currents in these seas move in one of the four directions, and are marked on the map in the following way: west-east `'<'`, east-west `'>'`, north-south `'v'` and south-north `'^'`. When the ducks are located on a cell with a current, they will move one cell in the direction of the current. Ocean currents in these seas are special, as they never lead outside of the map and they don't form vortexes (places where the ducks would move in circles if they followed the current).

Calm sea is marked by a dot `'.'`. If the currents bring the ducks to a cell with calm sea or back to the starting island, they won't be able to continue the voyage. The island that the ducks want to visit is marked by a letter `'x'`.

The ducks don't want to stop their beach party. They kindly ask you to tell them if it's possible for them to get to the other island, and if it is, in which direction should they start their voyage. Since one of the ducks gets very seasick, they ask you to choose the direction that will make the voyage as short as possible. If there are multiple directions that yield the same minimal travel time, you should choose the one that is alphabetically first.

Input

The first line contains integers r and s ($3 \leq r, s \leq 100$), the number of rows and columns of the map.

Each of the next r lines contains s characters from the set `'o<>v^.x'`, that represent the map of ocean currents. There will always be exactly one character `'o'` and exactly one character `'x'` on the map. The character `'o'` will never be located in the first or last row nor column.

Output

If the ducks can't reach the other island, print `:(`.

Otherwise, print `:)` in the first line. In the second line, print the start direction (`N` for north, `E` for east, `W` for west or `S` for south).

Scoring

In test cases worth 30 points the valid start direction will be unique if it exists.

Examples

input

```
6 6
..>>>v
.o^..v
.v.<.v
.>>^.v
.x<<<<
......
```

output

```
:)
E
```

input

```
5 5
v<<<<
>v.>^
v<.o.
>>v>v
..>>x
```

output

```
:)
S
```

input

```
3 3
x>.
.o^
^<.
```

output

```
:(
```

Clarification of the first two examples:

In the first example, if the ducks start their voyage in any direction but east, they will end up in calm sea and won't reach the other island.

In the second example, the ducks will reach the other island if they start by going north or south. They choose the south way, since it's shorter.

Task Bajka

Little Fabijan got bored with picture books, so he decided to read his first fairytale. Unfortunately, Fabijan often encounters a word that scares him. To overcome his fear, he will play a game he invented.

The scary word can be represented as an array of n lowercase letters. To start the game, Fabijan puts his finger on some position of the array and writes the letter from that position on a piece of paper. He then performs one of the following moves an arbitrary number of times:

- He moves the finger to a position that is one place to the left or to the right of the current position, if that position exists. Also, Fabijan will then write the letter from the new position on the paper, after the last written letter.
- He moves the finger to any position with the same letter as the current one. Fabijan will not write anyhing on the paper in this case.

It takes him $|x - y|$ seconds to move the finger from position x to position y.

Fabijan will overcome his fear of the word if, at the end of the game, his favourite word is written on the paper. He wants to finish the fairytale as soon as possible, so he asks you to tell him the minimum number of seconds it will take him to overcome his fear of the given scary word.

Input

The first line contains integers n and m $(1 \le n, m \le 300)$.

The second line contains n lowercase letters, the word that scares Fabijan.

The third line contains m lowercase letters, Fabijan's favourite word.

Output

Print the shortest possible time in seconds Fabijan needs to write his favourite word on the paper, or -1 if that is not possible.

Scoring

In test cases worth 20 points, letters in the word that scares Fabijan will be pairwise distinct.

Examples

input

```
2 2
wa
ac
```

output

```
-1
```

input

```
7 7
monolog
nogolom
```

output

```
10
```

input

```
14 5
niskoobrazovan
boook
```

output

```
5
```

Clarification of the third example:

Fabijan will first put his finger on position 7 and write down the letter `'b'`. He will then move the finger two times to the left, and each time write down the letter `'o'`. In the next step, he will move the finger to position 6 using the second type of move. Finally, he will again move the finger two times to the left, and write down the letters `'o'` and `'k'`. It took him five seconds in total, one second per move.

Task 3D Histogram

Mr. Malnar (over the phone): Listen, I had to put up some posters around Zagreb in the middle of the night. I stumbled upon a fence that was made out of some planks of varying heights, and I was thinking about how to calculate the largest area of a poster that I can fit on the fence. Wouldn't that make a nice COCI problem?

Associate: You were doing what?! Anyway, the problem is no good. It's a standard trick with a monotone queue, we even teach that to elementary school students on our camps.

Mr. Malnar: What if you twist it a bit, ask for the answer for every prefix or something like that, that should be hard enough.

Associate: That exact problem was featured last year in our CERC qualification contest. A tough one, it boils down to the *Harbingers* trick, but now everyone has seen it.

Mr. Malnar: Are you sure there is nothing we can do?

Associate: I think we have exhausted all problems with histograms. COCI 2010/2011 (Tabovi), COCI 2015/2016 (Poplava), COCI 2017/2018 (Krov), IOI selection test 2018 (Histogram)... You get the point.

Mr. Malnar: What if the histogram is three-dimensional?

Associate: Umm...

You are given a 3D histogram, that consists of n blocks that are placed next to each other. The i-th block is 1 meter wide, a_i meters tall and b_i meters long. In other words, from the front it looks like a histogram with bars of heights $a_1, a_2, \ldots a_n$, and from the top it looks like a histogram with bars of heights $b_1, b_2, \ldots b_n$.

Determine the **block with maximum volume** that can be placed inside the given 3D histogram. The sides of that block need to be parallel with the sides of the blocks that make up the 3D histogram.

Input

The first line contains the integer n from the task description.

The i-th of the following n lines contains integers a_i and b_i ($1 \leq a_i, b_i \leq 10^6$) from the task description.

Output

Print the volume in cubic meters.

Scoring

Subtask	Points	Constraints
1	20	$1 \leq n \leq 2000$
2	90	$1 \leq n \leq 200\ 000$

Examples

input

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

output

```
24
```

input

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

output

```
8
```

input

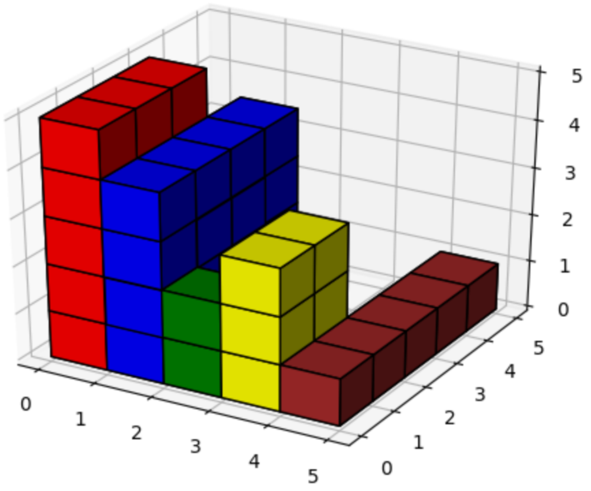
```
5
15 19
5 6
1 13
3 7
1 2
```

output

```
285
```

Clarification of the first example:

The figure below shows the 3D histogram from the first example. The largest block is obtained using parts of the first two blocks, and it is 2 meters wide, 4 meters tall and 3 meters long. The volume of the block is $2 \cdot 4 \cdot 3 = 24$ cubic meters.



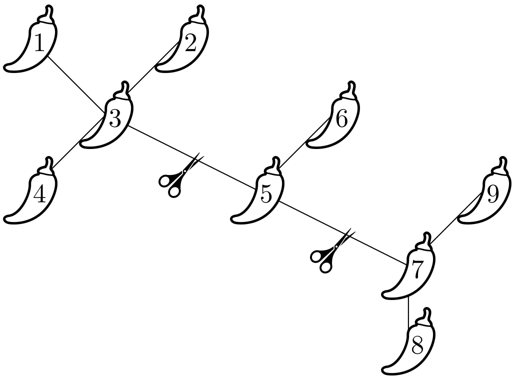
Task Papričice

Afrika paprika! – S.V.

After a tiring morning in the garden, Mr. Malnar decided to reward himself with dried hot peppers he grew himself.

He has n peppers, connected with $n-1$ pieces of string, so that every two peppers are connected by some series of strings. Formally, they form a *tree*.

Mr. Malnar will partake in three lunches today. For that purpose, he will **cut two strings** to get three smaller components, one for each lunch.



The tree from the third example along with the optimal cuts.

It's bad to make any lunch too spicy, so he will choose the cuts in a way that **minimises the difference between the size of the largest and the smallest component**. You need to determine the sought minimum difference.

Input

The first line contains an integer n, the number of peppers. The peppers are labeled by integers from 1 to n.

Each of the following $n-1$ lines contains two integers x and y ($1 \leq x, y \leq n$) – labels of peppers that are directly connected by a piece of string.

Output

Print the minimum possible difference of component sizes.

Scoring

Subtask	Points	Constraints
1	15	$3 \leq n \leq 200$
2	35	$3 \leq n \leq 2000$
3	60	$3 \leq n \leq 200\,000$

Examples

input

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

output

```
1
```

input

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

output

```
0
```

input

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

output

```
2
```

Clarification of examples:

In the first example, each of the possible three ways of cutting the strings yields one component with two peppers and two components with one pepper each. Therefore the answer is $2 - 1 = 1$.

In the second example, it's possible to get three components of the same size by cutting the string that connects peppers 1 and 3, and also 3 and 5, so the answer is 0.

Optimal cuts for the thirds example are shown on the figure in the statement. The sizes of the components are 4, 2 and 3, and the answer is $4 - 2 = 2$.

Task Tenis

Mirko is a big fan of tennis. Soon an important tournament with n players will take place. Mirko has spent years researching tennis players and has collected various statistics about the competitors. He has ranked their ability on three different courts: grass, clay and hard. More precisely, for each court he has determined the order (rank list) of the players, with the first player being the strongest, and the last being the weakest.

On this tournament, every player will play against every other player exactly once, so there will be $\frac{n(n-1)}{2}$ matches in total. A tennis match cannot end in a draw, and **the player who is stronger on the court the match is played on will win**. The organisers know that, so they have decided that each match should be played on the court for which the winner will be the strongest, i.e. have the best position in the corresponding rank list. If some courts are equal in that sense (the position of the winner of the match between players A and B would be the same, e.g. player A would win on court 1, and player B on court 2, and they are both ranked third on the corresponding court rank lists), they will choose the court for which the *loser* would have the best position. If the courts are still equal, the one with the smallest index is chosen.

Determine the outcome of this tournament: the number of matches played on each court and the number of wins for each player.

Input

The first line contains an integer n, the number of players. The players are labeled with integers from 1 to n.

The i-th of the following three lines contains a permutation of integers from 1 to n, the rank list of the players for the i-th surface, starting with the strongest player.

Output

In the first line, print the number of matches played on the first, second and third surface.

In the second line, print the number of matches won by each player from 1 to n.

Scoring

Subtask	Points	Constraints
1	35	$1 \leq n \leq 300$
2	15	$1 \leq n \leq 3000$
3	60	$1 \leq n \leq 100\ 000$

If your solution prints at least one of the lines correctly on each test case of a subtask, but it doesn't print both lines correctly on at least one test case, you will get half of the points for that subtask.

Examples

input

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

output

```
1 2 0
2 0 1
```

input

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

output

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

Clarification of the first example:

The match between players 1 and 2 is played on court 2, because the winner (player 1) has the best (first) position. For the match between playes 1 and 3, the winner has the same position on all three courts, but the loser has a better position on court 2. For the match between players 2 and 3, court 1 and 3 are equal, so the one with the smaller index is chosen (court 1).