Belgium Algorithm Contest

Belgium Algorithm Contest
Round 4 - 2018

Do not open before the start of the contest.

This page is not really blank.

Belgium Algorithm Contest

Problem A
Swipe patterns
Time limit: 1s

Have you ever wondered how many different patterns we can make on a 3×3 lock screen on a phone?

We model this with by using 9 points on the plane $(0,0)$, $(1,0)$, $(2,0)$, $(0,1)$, $(1,1)$, $(2,1)$, $(0,2)$, $(1,2)$, $(2,2)$ numbered from 1 to 9 in that order. A pattern will be a polyline such that:

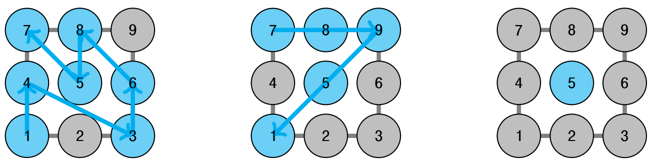
- Each segment of the polyline starts and ends at one of the 9 points.
- The polyline stays inside the square $[0,2] \times [0,2]$.
- Each point is visited at most one time.
- The polyline does not self-intersect.

Each pattern is identified by the order of the numbers of the points it visits. Two patterns are said to be different if this sequence is different. This means, in particular, that the same pattern performed in reverse order is counted as a new pattern. Note that in particular it is impossible to go in one segment from point 1 to 3 without passing by 2. Thus no sequence will have $(\ldots, 1, 3, \ldots)$.

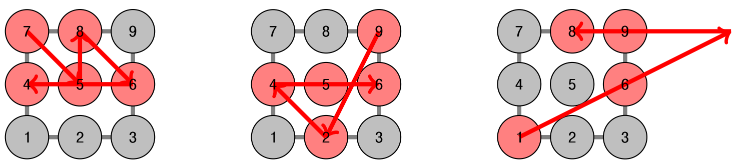
Your task is you count how many distinct patters exist.

Example:

The following figure illustrates four examples. The first two are valid patterns and their identifiers are shown below. The third pattern is invalid because 5 is used twice. The forth pattern is invalid because it self-intersects. Finally, the last one is invalid because it goes out of the bounds.



$id = (1,4,3,6,8,5,7)$ $id = (7,8,9,5,1)$ $id = (5)$



repeats 5 self-intersects out of bounds

Input

This problem has not input.

Output

The number of distinct valid pattern id's on a 3×3 lock screen (as described above).

Note:

You have to submit the code that you used to compute the answer not just one print with the answer. This will not be checked automatically by the system but the jury reserves itself the right to mark as incorrect any suspicious submission, even after the contest has ended.

Belgium Algorithm Contest

Problem B
Measuring islands
Time limit: 1s

You are part of a team hired to measure the coastline of an island. The plan is that the team will build a robot that is able to walk along the coast line and then, when the robot finishes its tour, you will have to deduce the area of the island based on the path followed by the robot.

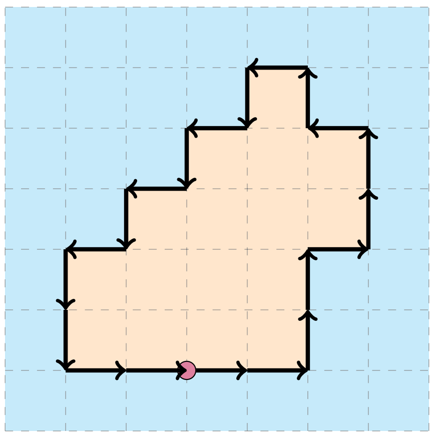
Fortunately the island has a very simple shape: its coast line is a polyline with each segment being either horizontal or vertical and having integer coordinates. The island has not holes.

The output of the robot is a sequence of characters R, L, U, D representing whether the robot took a step right, left, up or down.

Knowing the sequence of actions that the robot took, can you compute the island area?

Example:

Consider the island in the following figure. If the robot started at the purple dot and followed the arrows, then it will output the sequence `RRUURUULULDLDLDLDDRR`. From this sequence you have to deduce that the area of the island is 16.



Note that the robot needs to output one instruction for each step even if two consecutive steps go in the same direction. Also, you do not know where the robot starts, all you know is that the sequence of actions will bring it back to the same place while visiting the coastline of the island. Apart from the starting point which is visited exactly twice, no other point of the coastline was visited more than once by the robot.

Input

A single line with a string s over the alphabet RLUD giving the robot path.

The island coastline **does not** contain self intersections.

Constraints

1. $4 \le |s| \le 10^6$

Output

A single line with the area of the island.

Sample test cases

Input 1	Output 1
`RRUURUULULDLDLDLDDRR`	`16`

Input 2	Output 2
`ULDR`	`1`

Belgium Algorithm Contest

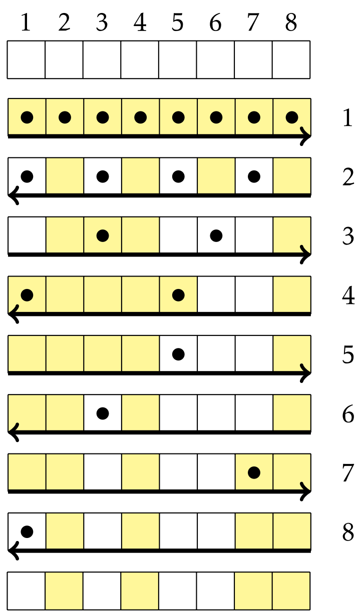
● Problem C
Lights
Time limit: 1s

There are n rooms along a very long hallway numbered from 1 to n. Initially the light in each room is turned off. As you are bored, you decided to occupy your time by walking back and forth n times (total, not n times in each direction). On the i-th walk you switch the light state of every i-th room. This means that in the first walk you switch each light, on the second walk you switch every two lights, and so on. Note that the first walk goes from left to right, on the second from right to left and so on.

Now the question is, after you are done, can you figure out the state of the light in some given room k?

Example:

Suppose that $n = 8$. The following figure illustrates what happens at each iteration. The squares denote the rooms: a white square represents a room with the light off and a yellow square represents a room where the light is on. The arrow at the bottom shows the walking direction and the dots represent the lights that were switched. The state of the lights that is represented is the state after that walk is finished. The first and last rows show the initial and final states respectively. The numbers on the top represent the room indexes and the numbers of the right the walk indexes.



Input

The first line of the input contains a single integer n giving the number of rooms.

The second line of the input contains a single integer k giving the room about which we want to know the final state.

Constraints

1. $1 \leq n \leq 10^{12}$
2. $1 \leq k \leq n$

Output

A single line with `on` if the light in room k is on at the end or `off` otherwise.

Sample test cases

Input 1	Output 1
8 4	on

Input 2	Output 2
8 5	off

Belgium Algorithm Contest

Problem D
Cheap network
Time limit: 1s

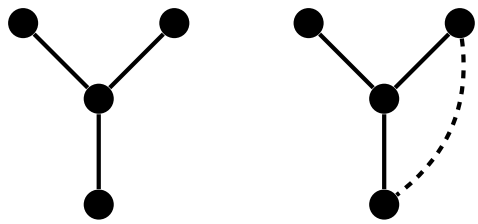
A minimal requirement in terms of connectivity for a computer network is to have at least one path between any two routers. You have been contacted by a network operator that used a network topology such that it contains **exactly** one path between any two routers. They have been receiving a lot of complaints from theirs users because whenever a link is down, some users cannot communicate with each other anymore.

As they are short in money, they would like to add only one new connection in the network so that the probability that a link failure disconnects the network is minimized.

Can you help them?

Example:

The following figure shows a network on the left and on the right one possible solution is provided. The probability that a link failure disconnects the network after this link is added is 1/4. Note that in this case any link that you add will achieve this.



initial network possible solution

Input

The first line contains a single integer n with the number of routers in the network.

Then follow $n-1$ lines describing the links in the network. Each such line contains two integers x and y giving that router x is connected to router y. Routers are numbered from 0 to $n-1$.

The network contains exactly one path between any to routers.

Constraints

1. $1 \leq n \leq 10^6$

2. $0 \leq x, y \leq n-1$.

Output

A single line with the probability that the network gets disconnected after a link failure if the new connection is placed so that that probability is minimal.

Your answer should have an absolute or relative error of at most 10^{-6}.

Sample test cases

The first sample input corresponds to the example above.

Input 1

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

Output 1

```
0.25
```

Input 2

```
3
0 1
1 2
```

Output 2

```
0
```

Belgium Algorithm Contest

Problem E
Exams
Time limit: 1s

You were hired to help scheduling the exams in your university.

You are given a list of exams each described by two integers: d, the day in which it occurs and s, the number of supervisors needed.

You are also given a list of supervisors. Each supervisor is described by an integer m giving the maximum number of exams he is willing to supervise and a list of days $a_1, \ldots, a_k$ on which he is available.

The goal is to assign exams to the supervisors so that each exam has exactly the needed number of supervisors assigned to it, no supervisor is assigned to an exam occurring on a day on which he is not available and no supervisor is assigned to more than he is willing to supervise. Also, a supervisor cannot be assigned to two exams occurring on the same day.

Example

Suppose that we have 3 supervisors and 4 exams as described in the tables below.

	m	a_i
Supervisor 1	2	0
Supervisor 2	1	0, 1
Supervisor 3	3	0, 1, 2

	d	s
Exam 1	0	1
Exam 2	0	2
Exam 3	1	1
Exam 4	2	1

One possible solution is to assign supervisors 1 and 2 to exam 2 and the third supervisor to exams 1, 3 and 4.

Input

The first line of the input contains 3 integers S, E and D representing the number of supervisors, exams and days, respectively.

The follow S groups of three lines describing the supervisors. The first line on each group contains the value m giving the maximum number of exams that the supervisor can be assigned to. The second line contains the number of available days k. Finally the third line contains k integers $a_1, \ldots, a_k$ giving the days on which the supervisor is available.

Then follow E lines each with two integers d and s giving the day of the exam and the number of supervisors needed.

Constraints

1. $1 \leq S \leq 50$
2. $1 \leq E \leq 50$
3. $1 \leq D \leq E$
4. $1 \leq m \leq E$
5. $1 \leq k \leq D$
6. $0 \leq a_i \leq D - 1$
7. $0 \leq d \leq D - 1$
8. $1 \leq s \leq S$

Output

A single line with `yes` if it is possible to assign the people to the exams and `no` otherwise.

Sample test cases

The first sample input corresponds to the example above.

Input 1	Output 1

Input 1:

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

Output 1:

```
yes
```

Input 2

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

Output 2

```
no
```

This page is not really blank.

This page is not really blank.