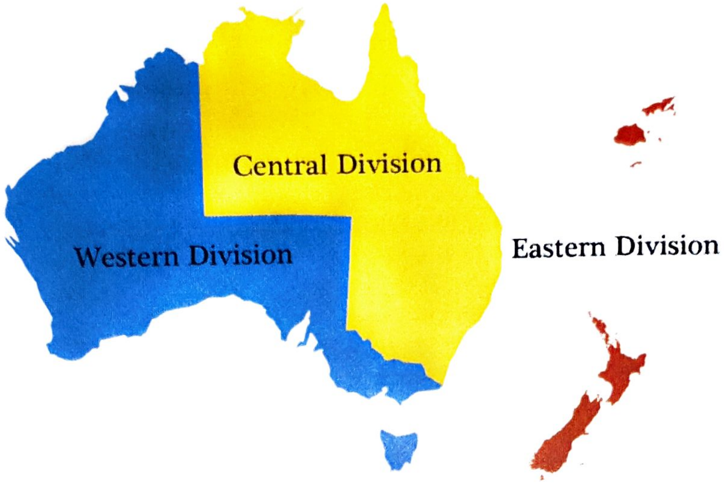


ICPC SOUTH PACIFIC DIVISIONALS

OCTOBER 20, 2018

Contest Problems

A: Australian vs American
B: Bombs Ahoy!
C: Cryptic Clues
D: Dubious Recording
E: Easiest Problem
F: Flow
G: God's Number
H: Holiday
I : Island of Love
J : Juice Machine
K: Krazy Taxi
L: Love Actually

This contest contains twelve problems over 26 pages. 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 $s_k < t_k$

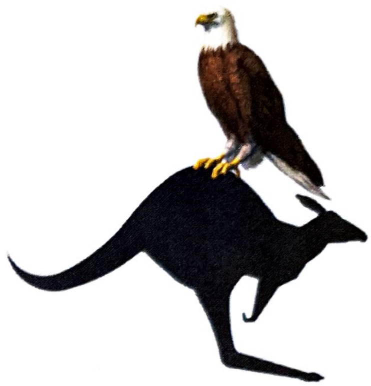
  or
- $s_i = t_i$ for all $1 \leq i \leq \min(n, \ell)$ and $n < \ell$.

Definition 3

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

Problem A
Australian vs American
Time limit: 1 second

You have just received a list of potential dates for the next South Pacific Regional Programming Contest. You would like to clear your schedule in anticipation, but then you realise that you are not sure what format the dates are in! All of the dates are written as xx/xx/xx, where each pair of digits represents either the day, the month or the year (the last two digits of a year are of the form 20YY).



Dates around the world are represented in different ways. For example, in Australia, 03/04/17 is read as April 3, 2017 (the day first, then the month, then the year). However, in the USA, the same string is interpreted as March 4, 2017 (the month first, then the day, then the year). In total, there are 6 interpretations of the order of the three numbers. Do not forget that for the dates in question, if the year is a multiple of four, then there is an extra day at the end of February (a *leap day*).

Given the list of dates, either determine which format they are in, determine that it is not possible to tell, or determine that they cannot all be in any consistent format.

Month	**Days**	**Month**	**Days**
(01) January	31	(07) July	31
(02) February	28 (29 in a Leap Year)	(08) August	31
(03) March	31	(09) September	30
(04) April	30	(10) October	31
(05) May	31	(11) November	30
(06) June	30	(12) December	31

Input

The input begins with an integer n ($1 \le n \le 100$), which is the number of dates.

The next n lines describe the dates. Each line contains a string of the form xx/xx/xx, where each x is a digit.

Output

If the list of dates is valid in exactly one interpretation, display the format as AA/BB/CC, where each of AA, BB, and CC are one of DD (day), MM (month), or YY (year). Otherwise, display UNSURE if multiple interpretations are valid or IMPOSSIBLE if the list of dates is invalid under any interpretation.

Sample Input 1

```
2
01/03/02
02/04/01
```

Sample Output 1

```
UNSURE
```

Sample Input 2

```
3
12/04/50
05/15/51
11/28/52
```

Sample Output 2

```
MM/DD/YY
```

Sample Input 3	Sample Output 3
3 20/05/01 07/17/12 03/04/49	IMPOSSIBLE

Sample Input 4	Sample Output 4
2 12/50/01 01/50/21	MM/YY/DD

Sample Input 5	Sample Output 5
4 01/27/02 01/05/03 32/13/09 02/01/12	YY/DD/MM

Problem B
Bombs Ahoy!
Time limit: 6 seconds

Georgie the mouse is sitting in a $N \times M$ grid, worriedly glancing at the sky. This is because he knows that K paint bombs will soon be dropped on the grid from above. Each paint bomb will splatter a rectangular section of the grid with paint.

Georgie does not want his beautiful fur to be painted either by stepping into an already painted square or remaining in a square when a bomb lands. Note that some of the grid squares may already be painted before any bomb has landed.

The K bombs will land in order. Just before each bomb lands, Georgie can move to a neighbouring grid square, or stay in the square he is in. Two squares are neighbouring if they share an edge or a corner. Note that Georgie can move at most once before the next bomb lands. Your task is to help Georgie navigate the grid so that he has no paint on his fur after all the bombs have landed. Georgie stops moving when the last bomb lands.

Input

The input starts with a line containing three integers N ($1 \leq N \leq 100\,000$), which is the number of grid rows, M ($1 \leq M \leq 100\,000$), which is the number of grid columns, and K ($1 \leq K \leq 100\,000$), which is the number of bombs that will be dropped. Furthermore, ($1 \leq N \cdot M \leq 500\,000$).

The next N lines describe the grid. Each line contains a single string of length M consisting only of the characters `S`, which is the starting position of Georgie, `X`, which denotes an already painted grid square, and `.`, which denotes an unpainted grid square.

The next K lines describe the bombs that will be dropped. Each line consists of four integers r_1, c_1, r_2, and c_2 ($1 \leq r_1 \leq r_2 \leq N$ and $1 \leq c_1 \leq c_2 \leq M$). Every grid square within the rectangle defined by these four points will be painted after the bomb is dropped. In other words, a grid square located at (r, c) will be painted if $r_1 \leq r \leq r_2$ and $c_1 \leq c \leq c_2$. Rows are numbered from top to bottom and columns from left to right

Output

If there is no way for Georgie to avoid getting paint on his fur, display -1.

If it is possible for Georgie to avoid getting paint on his fur, display the row and column of Georgie's final position. If there are multiple possible final positions, display the position with the smallest row index (breaking ties by selecting the smallest column index).

Sample Input 1	Sample Output 1
5 5 3 ..X.. ....S.. ....X 2 2 4 3 1 4 5 5 5 1 5 2	5 3

Sample Input 2

```
5 5 3
..X..
.....
..S..
....X
.....
2 2 4 3
1 3 4 5
5 2 5 5
```

Sample Output 2

```
-1
```

Problem C
Cryptic Clues
Time limit: 3 seconds

Cryptic crosswords contain a variety of clue types. For this problem, we consider *anagram* style clues. For example, `society that reconfigures mac` could be a clue with the solution `acm`.

An anagram is a permutation of a letter sequence. The sequence `mac` can be permuted into `acm`, for example. Letters that occur more than once are considered indistinguishable. A letter sequence is not considered to be an anagram of itself, so `aab` has only two anagrams: `aba` and `baa`.

In this problem, letters in the clue should be permuted to form the solution. These letters must consist of one or more consecutive whole words in the clue, with spaces between words being dropped. For example, in the clue `machine that reuses cut eprom`, the consecutive words `cut eprom` would become `cuteprom` which means a possible solution to the clue is `computer` (`cuteprom` is an anagram of `computer`). Note that `cuteprom` is not a possible solution.

You have a clue. The clue is a sequence of words separated by spaces. You also have a list of candidate solutions. Each of these is a distinct word. Your task is to find the candidate solution corresponding to the clue. This is guaranteed to exist and to be unique.

Input

The input starts with a line containing a single string, which is the clue. The clue contains only lowercase letters with a single space separating words in the clue. A word is a sequence of one or more letters. The total number of letters in the clue is between 2 and 100, inclusive.

The next line contains a single integer W ($1 \le W \le 100$), which is the number of candidate solutions.

The next W lines describe the candidate solutions. Each line contains a single word which consists of between 2 and 100 lowercase letters inclusive.

Output

Display the candidate solution matching the clue.

Sample Input 1

```
south pacific icpc
5
icpcsouth
pacificicpc
icpcpacific
southpacific
sppc
```

Sample Output 1

```
icpcpacific
```

Sample Input 2

```
society that reconfigures mac
1
acm
```

Sample Output 2

```
acm
```

Sample Input 3

```
hitperson turning off re offer of ref
2
oofer
offer
```

Sample Output 3

```
offer
```

Problem D
Dubious Recordings
Time limit: 2 seconds

The Australian Federal Police have intercepted many potentially dubious recordings. These were obtained by circumventing the encryption schemes used to secure online messaging services. Usually, this would be almost impossible. However, in Australia, a new law was created that allows circumvention of encryption. As we all know, the laws of mathematics are very commendable, but the only law that applies in Australia is the law of Australia. The recordings all appear to be sound files of some sort. The police would like to determine which recordings are *dubious*.

A recording is a sequence of samples. Each sample can be represented as a number. A recording is considered dubious if the average of the sums of all (non-empty) contiguous subsequences is too high. This average is called the *overall average* of the recording. This may seem arbitrary, but, Jane, the forensics expert, insists that she needs to know the average. Formally, consider a recording R. We use R_i to denote the i^{th} sample in R. Let $S(l, r) = \sum_{i=l}^{r} R_i$. The overall average is the average of $S(l, r)$ over all l and r such that $l \leq r$.

For example, Consider the sequence 1, 2, 3. This has 6 ranges to consider (`1`, `1 2`, `1 2 3`, `2`, `2 3`, `3`). The sums of each of these are 1, 3, 6, 2, 5, 3. Thus, the average is $\frac{20}{6}$. This can be simplified to $\frac{10}{3}$.

What is the overall average?

Input

The first line of input contains a single integer N ($1 \leq N \leq 200\,000$), the length of the recording.

The second line describes the recording. This line contains N integers $a_1, a_2, \ldots, a_N$ ($1 \leq a_i \leq 1\,000$), where a_i is the i^{th} sample.

Output

Display the overall average of the recording as a fraction in its reduced form. A fraction is in reduced form if its numerator and denominator contain no common factors other than 1.

Sample Input 1

```
1
5
```

Sample Output 1

```
5/1
```

Sample Input 2

```
3
1 2 3
```

Sample Output 2

```
10/3
```

Problem E
Easiest Problem
Time limit: 1 second

When doing ICPC contests, one of the most important skills is correctly determining the difficulty of each problem quickly. To get the best placement in an ICPC contest, it is in a team's best interest to solve the easiest problem first and work on the harder problems later.

To ensure that they find the easiest problem, all three members of Team Merlin will read all of the problems and rank each one in terms of difficulty. Each ranking is an integer between 1 and 100, where 1 is the easiest and 100 is the hardest. A problem's *difficulty score* is the average of the three team members' rankings. The problem with the lowest difficulty score is deemed to be the *easiest problem*. In the case of a tie, the problem that received the lower ranking from the team captain is the easiest problem. It is guaranteed that the captain will not give two problems the same rank.

What is the easiest problem? Sample 2 is not an indication of actual difficulty.

Input

The first line of the input contains a single integer P ($1 \leq P \leq 15$), which is the number of problems in the contest.

The next P lines describe the problems. Each line starts with a string, which is the name of the problem. Then follow three integers a, b, c ($1 \leq a, b, c \leq 100$), which are the three rankings of the members of Team Merlin. The team captain's ranking is a. The problem's name uses only lowercase and uppercase letters and consists of between 1 and 50 characters inclusive.

The P problem names will be distinct. Problem names are case-sensitive.

Output

Display the name of the easiest problem.

Sample Input 1

```
2
AnEasyProblem 5 3 7
AHardProblem 92 81 97
```

Sample Output 1

```
AnEasyProblem
```

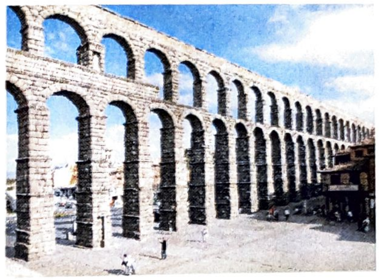
Sample Input 2

```
12
AustralianVsAmerican 10 10 10
BombsAhoy 20 20 20
CrypticClues 25 25 20
DubiousRecording 35 23 15
EasiestProblem 1 1 1
Flow 30 30 30
GodsNumber 70 75 80
Holiday 34 15 88
IslandOfLove 100 100 100
JuiceMachine 50 12 32
KrazyTaxi 23 30 30
LoveActually 99 99 99
```

Sample Output 2

```
EasiestProblem
```

Problem F
Flow
Time limit: 2 seconds



Maximus Aquius, the Roman aqueduct themed supervillain, is trying to irrigate the streets of Rome. He plans to do this by building a network of aqueducts that direct water from the surrounding countryside to the city. There are a number of water sources around Rome, each producing a fixed volume of water each hour. Each source of water has some altitude, and water can only flow downhill. If a water source has a lower altitude than the city itself, it cannot be used to irrigate the city.

Maximus Aquius, having no faith in your ability to solve complex problems, has hired a more talented computer scientist to compute the set of additional aqueducts he needs to build. Having now decided on his plan, he needs to figure out the logistics necessary to build the network.

Your task is to determine how many labourers must be hired to construct the aqueducts to ensure they are completed before Caesar returns from Gaul. Each labourer can build exactly 1 cubit of aqueduct each day. Multiple labourers can work on the same aqueduct at the same time and multiple aqueducts can be worked on at the same time, but a worker can only work on a single aqueduct on any given day. Given the length of each of the aqueducts (in cubits), what is the minimum number of labourers that must be employed to build all aqueducts before Caesar returns?

Input

The first line of input contains two integers m ($1 \leq m \leq 100\,000$), which is the number of aqueducts to be built, and n ($1 \leq n \leq 100\,000$), which is the number of days until Caesar returns.

The next m lines describe the aqueducts. Each line contains a single integer ℓ ($1 \leq \ell \leq 1\,000$), which is the length of this aqueduct in cubits.

Output

Display the minimum number of labourers required.

Sample Input 1	Sample Output 1
5 10 2 2 2 2 2	1

Sample Input 2	Sample Output 2
6 16 18 3 5 4 1 1	2

Problem G
God's Number
Time limit: 1 second

The Rubik's Cube is a $3 \times 3 \times 3$ cube puzzle. There are six colours and when solved, the nine squares on each face have the same colour. See Figure ??.

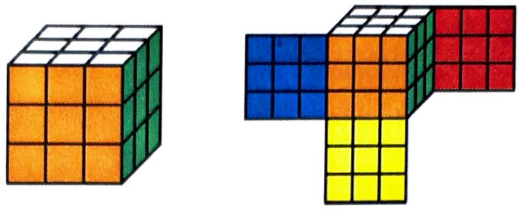
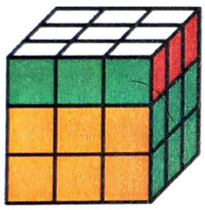
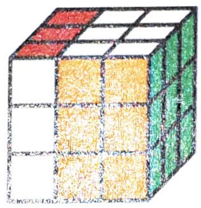


Figure G.1: A solved Rubik's Cube.

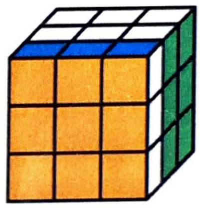
Each of the six faces can be rotated either clockwise or counterclockwise, which rotates the colours on that face as well as alters one row on the four neighbouring faces. Here are the six clockwise rotations by $90°$:



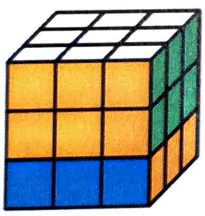
U (Up)



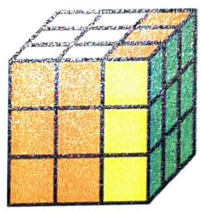
L (Left)



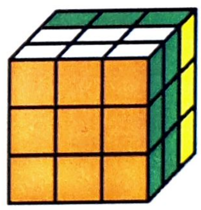
F (Front)



D (Down)



R (Right)



B (Back)

In 2010, it was proven that any Rubik's Cube configuration can be solved in at most 20 moves (one *move* here is rotating one face by either $90°$, $180°$ or $270°$ clockwise). In 2014, it was proven that if you were only allowed rotations of $90°$ or $270°$ (clockwise), then any Rubik's Cube configuration can be solved in at most 26 moves.

Jacob is attempting to tackle the final case: only allowing rotations of $90°$ (clockwise). He has been working towards a solution for several days, but cannot seem to even solve the Rubik's Cube! Thankfully, he started with a solved Rubik's Cube and he has kept track of the moves that he has made so far. Given this information, help solve the Rubik's Cube for him using only rotations of $90°$ (clockwise). You *do not* need to solve the Rubik's Cube in the minimum number of moves.

Input

The input consists of a single line containing a single string, which is the sequence of moves that Jacob has performed on the Rubik's Cube. The string consists of only B, D, F, L, R and U. The Rubik's Cube starts solved, then the sequence of moves are applied, in order. The length of the string is between 1 and 50, inclusive.

Output

Display a sequence of moves that solves the Rubik's Cube. Your solution does not need to be optimal. The sequence of moves must be in the same format as the input. Your sequence can be of any length between 1 and 200, inclusive.

Sample Input 1

```
DUDUDUD
```

Sample Output 1

```
U
```

Sample Input 2

```
FF
```

Sample Output 2

```
FF
```

Sample Input 3

```
URRFBRBBRUULBBRUUUDDDRRFRRRLBBUUFF
```

Sample Output 3

```
URRFBRBBRUULBBRUUUDDDRRFRRRLBBUUFF
```

Sample Input 4

```
FFFF
```

Sample Output 4

```
LLLL
```

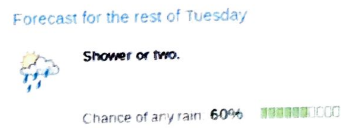
Sample Input 5

```
FB
```

Sample Output 5

```
FFFBBB
```

Problem H
Holiday
Time limit: 3 seconds



Your friend, Anna, from America is coming to visit you in Melbourne, Australia. "I hope the weather is nice," she says unaware of what Melbourne has in store for her. She has identified n consecutive days during which she might visit. She does not yet know when she will arrive or how long she will stay for. To prove to her how bad Melbourne's weather is you want to make sure she experiences every possible temperature in the n days.

It has been decided that you will pick the number of days d that Anna will stay for. Anna does not yet know what day she will arrive. However, she will stay for exactly d days when she does arrive (she will never arrive too late to stay for d days). You want to determine the minimum d such that no matter which day Anna arrives, she will experience all of the different temperatures in the n day span. That is, what is the smallest value of d such that no matter which of the first $n - d + 1$ days Anna arrives, she will experience every different temperature at some point during her d day trip?

Input

The first line contains an integer n ($1 \leq n \leq 200\,000$), which is the number of days.

The second line describes the temperatures. This line contains n integers $t_1, t_2, \ldots, t_n$ ($0 \leq t_i \leq 10^9$), which are the temperatures on each day in order.

Output

Display the minimum number of days Anna must stay for.

Sample Input 1
```
5
40 10 40 10 40
```
Sample Output 1
```
2
```

Sample Input 2
```
5
40 40 40 40 40
```
Sample Output 2
```
1
```

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

Sample Input 4
```
6
10 40 20 30 50 4
```
Sample Output 4
```
6
```

Problem I
Island of Love
Time limit: 4 seconds

Island of Love is an exciting new game show. On this show, many of the contestants start as friends. The show is quite a high-pressure situation. It involves long bouts of relaxing by the pool, drinking lots of beer, gossiping, and strenuous arguing. Since it is such a stressful situation, people stop being friends as the season progresses. All friendships are bidirectional. That is, if person X is friends with person Y, then person Y is also friends with person X. Also, as the show goes on, people never become friends, they only ever stop being friends (yes, it is depressing).

Two people are on *speaking terms* if they are friends or if there is some sequence of friendships that connect the two people. For example, if person X is only friends with person Y and person Y is friends with person Z, then person X and person Z are on speaking terms. However, if person Y and person Z end their friendship, then person X and person Z are no longer on speaking terms since there is no sequence of friendships connecting them.

Your job is to process some friendship ending events, as well as determine if two people are on speaking terms at certain points throughout the season. All the events and queries will be in the order of the time that they occur. You can assume they all happen at distinct times. A friendship ending event results in two people no longer being friends.

Input

The first line of input contains three integers N ($1 \leq N \leq 10^5$), which is the number of people on the show, F ($0 \leq F \leq 10^5$), which is the number of distinct pairs of friends when the show begins, and Q ($0 \leq Q \leq 10^5$), which is the number of events and queries to process.

The next F lines describe the friendships. Each of these lines contains two integers X and Y ($1 \leq X < Y \leq N$), which denote that person X is friends with person Y. It is guaranteed that all pairs are distinct.

The next Q lines describe the friendship ending events and queries in the order they occurred. Each of these lines contains three items. The line starts with a single character t (t is either `E` or `S`). Then follow two integers X and Y ($1 \leq X < Y \leq N$). If t is `E`, then there was a friendship ending event between person X and person Y (it is guaranteed that they were friends before this event). If t is `S`, then you must determine if person X and Y are on speaking terms at this point in time.

Output

For each query in the given order, display if the two people are on speaking terms at that given time.

Sample Input 1	Sample Output 1
3 3 4 1 2 1 3 2 3 E 1 2 S 1 2 E 1 3 S 1 2	YES NO

Sample Input 2	Sample Output 2
4 4 5 1 2 1 3 2 3 3 4 E 1 2 S 1 4 E 2 3 S 1 4 S 2 4	YES YES NO

Problem J
Juice Machine
Time limit: 2 seconds

Constantina is trying to buy some juice from a vending machine in her home country of Ayeseepeesee. She has to pay for the juice using coins. In this country, they have a frustrating coin system. While most countries have coins which are a certain number of cents (that is, a fraction in the form $\frac{x}{100}$), Ayeseepeesee takes this further and uses fractions with a different denominator (for example, $\frac{1}{3}$).

Assuming that Constantina has an unlimited supply of several different types of coins, what is the minimum number of coins she must use to pay for her juice? She must pay the exact amount for the juice. Note that we are interested in the minimum number of coins used, not the minimum number of coin types used.

Input

The input starts with a line containing two integers N ($1 \leq N \leq 50$), which is the cost of the juice, and C ($1 \leq C \leq 10$), which is the number of coin types Constantina has.

The next C lines describe the coin types she has. Each of these lines contains two integers n and d ($1 \leq n, d \leq 10$), which denote that Constantina has an unlimited supply of coins with value $\frac{n}{d}$.

Output

Display the minimum number of coins needed to pay the exact amount for the juice. If it is impossible to pay for the juice, display `-1`.

Sample Input 1	Sample Output 1
3 2 8 7 3 7	7

Sample Input 2	Sample Output 2
1 1 2 3	-1

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

Problem K
Krazy Taxi
Time limit: 1 second

There have been several complaints about a taxi that has been driving the wrong direction down one-way streets. Jennifer, a police officer, is responding to the complaints and has pulls the suspected taxi over.

After questioning the driver, Jennifer has determined that the driver started his night at his house and has travelled to the current location, but he claims that he has not done anything illegal. Jennifer goes back to her car and looks at the map:

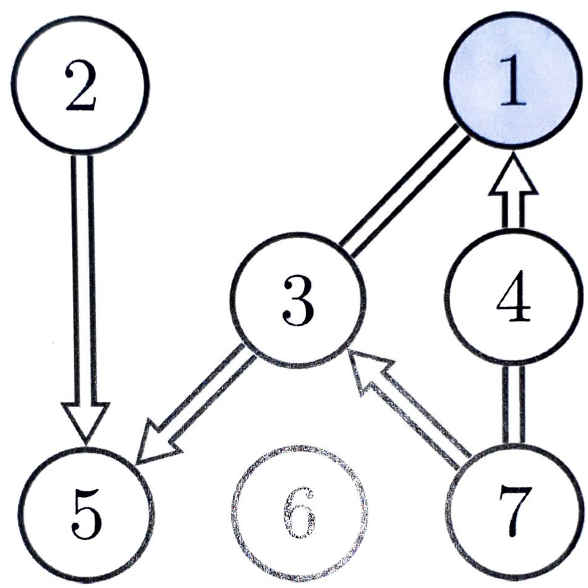


Figure K.1: Edges with arrows depict one-way streets. Edges without arrows depict two-way streets.

The driver's house is labeled as 1 on the map. Since they are currently on location 4, Jennifer can be sure that the driver went the wrong way down a one-way street (either 1-to-4 or 3-to-7). So Jennifer gives the driver a ticket. The next night, Jennifer pulls over the same driver at location 5. Since it is possible to get to location 5 via two-way streets and one-way streets in the correct direction, she cannot give him a ticket.

Since this is going to happen again and again, help Jennifer by writing a program that shows if she should give a ticket or if it is impossible to get to a location even by using one-way roads in the wrong direction from the driver's house.

Input

The first line contains three integers n ($1 \leq n \leq 1\,000$), which is the number of locations, x ($0 \leq x \leq 1\,000$), which is the number of two-way streets, and y ($0 \leq y \leq 1\,000$), which is the number of one-way streets.

The next x lines describe the two-way streets. Each of these lines contains two integers u ($1 \leq u \leq n$) and v ($1 \leq v \leq n$), which are the two end-points of the street.

The next y lines describe the one-way streets. Each of these lines contains two integers u ($1 \leq u \leq n$) and v ($1 \leq v \leq n$), which are the two end-points of the streets. The street goes from u to v.

It is possible that there are multiple streets between a pair of locations, but all streets connect two distinct locations. The locations are numbered 1 to n and the driver's house is location 1.

Output

For each of the locations in ascending order in order display whether or not Jennifer should give a ticket or if it is impossible to get there.

Sample Input 1

```
5 0 2
3 2
4 1
```

Sample Output 1

```
No Ticket
Impossible
Impossible
Ticket
Impossible
```

Sample Input 2

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

Sample Output 2

```
No Ticket
Ticket
No Ticket
Ticket
No Ticket
Impossible
Ticket
```

Sample Input 3

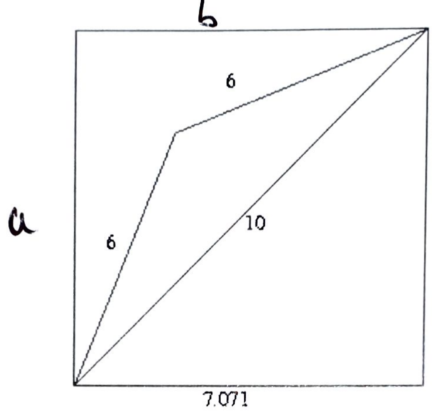
```
2 0 0
```

Sample Output 3

```
No Ticket
Impossible
```

Problem L
Love Actually
Time limit: 1 second

For Christmas this year, Steve has decided to give his lovely wife her favourite shape, a triangle, framed inside of her second favourite shape, a square. The triangle may be arbitrarily rotated and the entire triangle must fit inside of the square picture frame.



$a^2 + b^2 = c^2$

$2a^2 = c^2$

$a = \sqrt{\frac{c^2}{2}}$

Steve has already purchased the triangle and wants to know the smallest size of a square picture frame that can fit the triangle.

Input

The input consists of a single line containing 3 integers a, b and c ($1 \le a \le b \le c \le 10\,000$), which are the side lengths of the triangle. It is guaranteed that these values form a valid triangle ($a + b > c$).

Output

Display the minimum side length of a square that the triangle can fit in, with an absolute or relative error of at most 10^{-9}.

Sample Input 1	Sample Output 1
`6 6 10`	`7.071067811865475`

Sample Input 2	Sample Output 2
`3 4 5`	`3.880570000581328`