培正喇沙編程挑戰賽
LA SALLE – PUI CHING PROGRAMMING CHALLENGE

第一屆 · 2016

2016 年 8 月 22 日 (星期一)

香港培正中學

題號		名稱	時間限制	記憶體限制
A		Shuttle Bus	2 秒	256 MB
B		Salt Trading	1 秒	64 MB
C		Annoying Mathematics	2 秒	64 MB
D		Archery	1 秒	64 MB
E		Bacteria Experiment	2 秒	256 MB
F		Anniversaries	1 秒	64 MB
G		Monorail	1 秒	64 MB
H		Pokemon GO	2 秒	256 MB
I		RNG	1 秒	64 MB
J		Posters	1 秒	64 MB
K		Lattice Points	1 秒	64 MB
L		Textbook Game	2 秒	128 MB

Pascal	`ppcx64-3.0.0 -O2 -Sg -v0 -dEVAL -XS`
C	`gcc-5.3.1 -static -Wno-unused-result -DEVAL -lm -s -O2`
C++	`g++-5.3.1 -static -std=c++11 -Wno-unused-result -DEVAL -lm -s -O2`

編繹時間限制 10 秒　編繹記憶體限制 512 MB

作者　韓文軒 潘力行 董鑫泉 黃敏恆

This page is intentionally left blank.

A Shuttle Bus

Time Limit: 2 seconds
Memory Limit: 256 MB
Input: standard input
Output: standard output

Alex is a driver of a shuttle bus whose working duty is to drive around Byteland and let the tourists do sightseeing there.

The territory of Byteland is strange which can be represent by an grid with exactly 2 rows and N columns. There are M churches on some cells in Byteland where sightseeing there are forbidden. On the other hand, there is an attraction in each of the remaining cells.

On each day, Alex drives the shuttle bus from the frontier of Byteland, which is the top-left corner of the $2 \times N$ grid. The shuttle bus can travel from one cell to its adjacent cells which have a common side with it each time. Alex will drive the shuttle bus to visit all attractions. Undoubtedly, he cannot drive into the cells where the churches are located.

Alex does not want to make his tourists bored, so he hopes to visit all attractions, except the churches, exactly once. The tour can end in any cell. Given the length of the grid and the positions of the churches, determine whether Alex can do so successfully.

Input

The first line of contains 2 integers N, M, representing the length of the grid of Byteland and the number of churches there. ($1 \le N \le 10^9$, $1 \le M \le 5000$)

The following M lines contains 2 integers r_i, c_i, representing the position of the i^{th} church. ($1 \le r_i \le 2$, $1 \le c_i \le N$)

The positions of the churches are distinct and no church will be located at the top-left corner of the grid.

Output

Please output `Yes` if Alex can visit all attractions except the churches exactly once and output `No` otherwise.

Examples

input

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

output

```
Yes
```

input

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

output

```
No
```

input

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

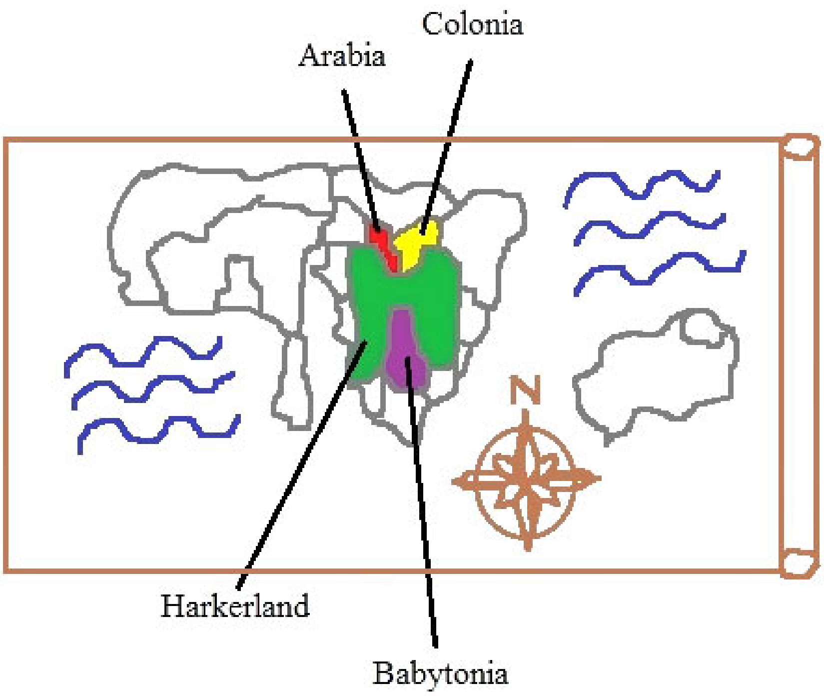
output

```
No
```

B Salt Trading

Time Limit: 1 second
Memory Limit: 64 MB
Input: standard input
Output: standard output

In the ancient kingdom by the name of Harkerland, where salt is aplenty, people often carry large bags of salt to neighbouring kingdoms for trading. In Harkerland, one kilohark (kh) of salt is worth exactly H$1. Harkerland is so vast, that it borders more than 10 kingdoms. Of these kingdoms, however, most are coastal and salt would not sell a good price there. Only three kingdoms are inland and merchants there would certainly purchase salt at a rate higher than H$1/kh.



The three inland kingdoms, Arabia, Babytonia, and Colonia, are as labelled in the map above.

- Arabian merchants are willing to buy salt at a rate H$$A$/kh.
- Babytonian merchants are willing to buy salt at a rate H$$B$/kh.
- Colonian merchants are willing to buy salt at a rate H$$C$/kh.

Here, A, B, C are real numbers larger than 1, given with exactly two decimal places.

RB is a millionaire and a bit of a celebrity in Harkerland. Recently, he has successfully traded ten copies of his autographs for 100 kiloharks of salt. Despite RB's excessive liking for salt, his salt warehouse is already full and cannot take even an extra grain of salt. Therefore, RB decides to trade salt for gold.

RB wants to send three traders, one to Arabia, one to Babytonia, and one to Colonia, each carrying a portion of RB's salt. RB believes that 3 is a good number, hence exactly three traders are to be dispatched, even though some of them may end up carrying no salt at all. In addition, RB believes that dreams foretell future events. Last night, he dreamed that two of his traders would be robbed on the way, and hence only one trader would return with money.

Surely RB wants to maximize his profit, but being a cautious person, he wants first to ensure that he will earn a positive amount of money in this trade. To be precise, RB wants to distribute his 100 kiloharks of salt among the three traders in a way, such that if exactly one trader is able to return, *no matter which one*, that trader would return with more than H$100, the total value of the salt RB originally has.

Given the values of A, B, and C, your task is to determine whether RB can earn money in the worst case by suitable allocation of his 100 kiloharks of salt.

Here are several assumptions related to the problem:

- The traders receive no monetary reward for their hard work — for them, being able to work for RB is itself their single biggest reward.
- Salt can be measured with arbitrary precision. For example, RB can let one of his traders carry $\frac{100}{7}$ kh, π kh, or 10^{-6} kh of salt.
- Harkerland currency can take any real value.
- Nothing will be lost during transport, except due to robbery.

Input

The first and only line of input consists of three space-separated real numbers A, B, and C, the selling prices of 1kh of salt in Arabia, Babytonia, and Colonia, respectively. Recall that A, B, C are real numbers larger than 1, given with exactly two decimal places. ($1 < A, B, C \leq 1000$)

Output

If RB can earn money in the worst case by suitable allocation of his 100 kiloharks of salt, output `Yes`. Otherwise, output `No`.

Examples

input	output
`3.00 3.00 3.00`	`No`

input	output
`3.00 4.00 2.95`	`Yes`

Note

For sample test 1, the best RB can do is to distribute $\frac{100}{3}$ kh of salt to each of the three traders. In this case, any returning trader would carry H$$3.00 \times \frac{100}{3}$ = H$$100$.

For sample test 2, RB can, for instance, give 34 kiloharks of salt to the trader heading to Arabia, 25.543 kiloharks of salt to the trader heading to Babytonia, and 40.457 kiloharks of salt to the trader heading to Colonia.

- If the Arabia-bound trader returns, RB would get H$$3.00 \times 34$ = H$$102$.
- If the Babytonia-bound trader returns, RB would get H$$4.00 \times 25.543$ = H$$102.172$.
- If the Colonia-bound trader returns, RB would get H$$2.95 \times 40.457$ = H$$119.34815$.

In the worst case that the Arabia-bound trader returns, RB could still earn H$$2$.

C Annoying Mathematics

Time Limit: 2 seconds
Memory Limit: 64 MB
Input: standard input
Output: standard output

Dr. Jones is a professor in Byteland Academy who always challenges his students with interesting mathematical problems about constructing sequences.

Today, Dr. Jones takes out R cards with $1, 2, ..., R$ written on them respectively. Then, he asks his student, Alex, to pick exactly N cards out of R cards in a way such that the lowest common multiple of the N numbers written on the chosen cards is equal to K.

For example, let $N = 3$, $R = 8$ and $K = 12$. The subsets $\{1, 4, 3\}$ and $\{2, 3, 4\}$ are examples of valid answer whereas subsets $\{2, 3, 6\}$ and $\{1, 6, 12\}$ are not.

As Alex hate mathematics, he is asking for your help. Please help him to find a valid subset. If there are more than one valid subsets, you can output anyone of them.

Input

The first and the only line contains three integers, N, R, K.

$1 \leq N \leq 10^5$, $1 \leq R, K \leq 10^9$

Output

Output N space-separated integers in one line, representing the subset of cards satisfying Dr. Jones' request. If there are more than one arrangement, output any one of them. You may output the numbers in any order.

If there is no arrangement satisfying Dr. Jones' request, output `-1`.

Examples

input	output
`4 8 8`	`8 1 2 4`
`5 6 30`	`5 6 1 2 3`
`1 12 13`	`-1`

This page is intentionally left blank.

D Archery

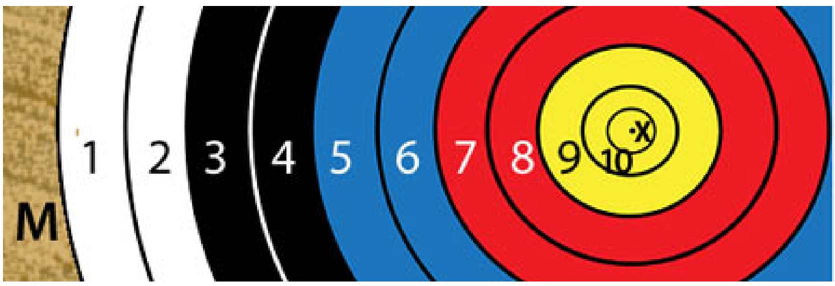
Time Limit: 1 second
Memory Limit: 64 MB
Input: standard input
Output: standard output

The Idol Star Athletic Championship 2016 archery elimination round has just finished! Help the judge by writing a program to read the scores of the two contestants, Yuju and Yerin, and determine who is going to advance.

Yuju and Yerin, *Idol Star Athletic Championship 2016*

Instead of 72, assume that the round is determined by N arrows. For an arrow, the score written on the scorecard can be one of the following:

- `M` for Miss (0 points)
- `1` to `9` (1 to 9 points respectively)
- `10` for Outer 10 (10 points)
- `X` for Inner 10 (10 points)



At the end of N arrows, the points are summed up to compute the total. The contestant with the higher total would advance. In case of ties, the contestant with more `10`s (`X` included) would advance. If still tied, the contestant with more `X`s would advance. If still tied, a shoot-off (extra arrows) would be required.

Input

The first line contains an integer N, the number of arrows shot by each contestant. ($1 \le N \le 100$)

The next N lines contain the scores of the arrows shot by Yuju.

The next N lines contain the scores of the arrows shot by Yerin.

Output

If Yuju advances according to the rules above, output `Yuju`. If Yerin advances, output `Yerin`. If shoot-off is required, output `Shoot-off`.

Examples

input

```
3
9
10
X
X
8
X
```

output

```
Yuju
```

input

```
2
4
1
M
6
```

output

```
Yerin
```

input

```
4
X
10
8
7
9
X
10
6
```

output

```
Shoot-off
```

Note

In the first sample, Yuju scored 29 points while Yerin scored 28 points.

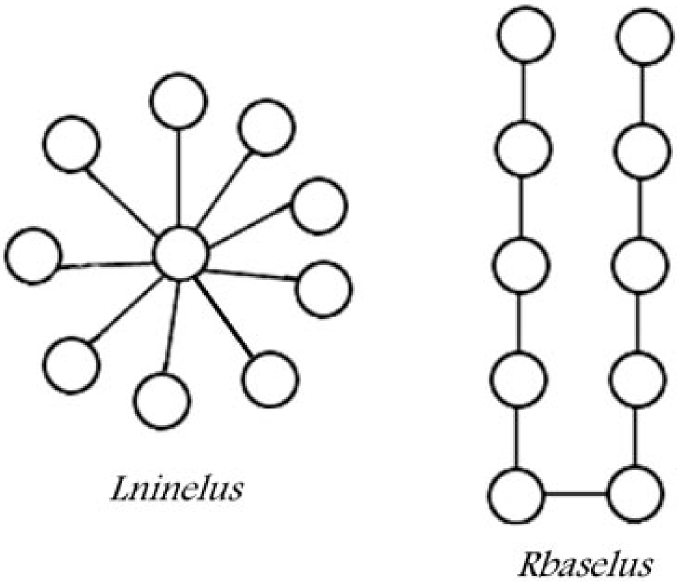
In the second sample, Yuju scored 5 points while Yerin scored 6 points.

In the third sample, both contestants scored 35 points, have two `10`s, and one `X`. Therefore, shoot-off is required.

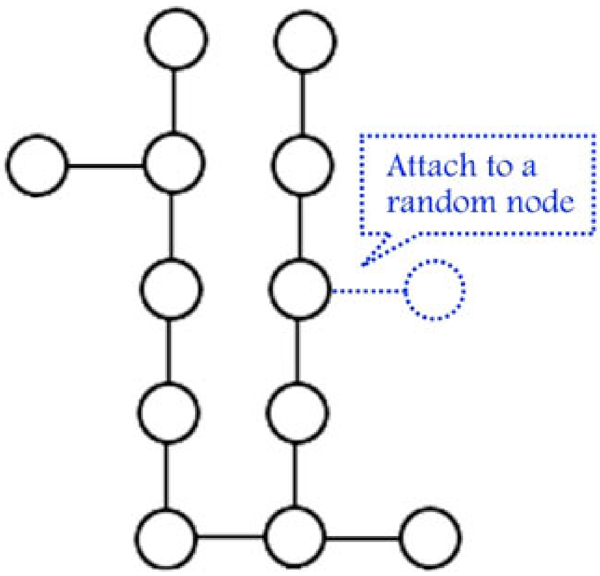
E Bacteria Experiment

Time Limit: 2 seconds
Memory Limit: 256 MB
Input: standard input
Output: standard output

Jason is performing a bacteria growing experiment. There are two types of bacteria involved, the *Lninelus* and *Rbaselus*. Each bacteria have a fixed seed with 10 cells; we can also view it as a tree with 10 nodes in terms of graph theory:



He is doing the experiment on T growing plates, where each plate either started with a Lninelus seed or a Rbaselus seed. These two bacteria have a special property: every hour, a new cell (node) will grow and attach to an existing cell (node) with **uniform probability**. As a result, this maintains the tree structure.



Jason let the bacteria grow for n - 10 hours. Now, each plate is a tree with n nodes. But, he realized he forgot to label the plates! Given the structure of the tree for each plate, can you determine which seed it is grown from? Your solution will be accepted if the accuracy is at least 95%.

Input

The first line contains two integers T and n ($900 \le T \le 1000$, $n = 1000$).

The following T lines each describe an unrooted tree in $n - 1$ integers: $p_1, p_2, \ldots, p_{n-1}$. There is an edge between node i and p_i ($0 \le p_i \le n - 1$ and $p_i \ne i$). The nodes are labelled from 0 to $n - 1$ in each tree. The labels are shuffled randomly and do not indicate the growth order.

Output

Output exactly T lines: for each tree in order of the input, output `L` if you think it is Lninelus; or `R` if you think it is Rbaselus.

Example

input

```
3 15
0 1 2 3 4 5 6 7 8 9 10 11 12 13
0 0 0 0 0 0 1 0 0 0 0 0 3 0
14 0 0 2 4 5 6 7 8 9 10 11 12 13
```

output

```
R
L
R
```

Note

The sample input does not satisfy the constraints of T and n; it serves to demonstrate the format.

F Anniversaries

Time Limit: 1 second
Memory Limit: 64 MB
Input: standard input
Output: standard output

Today is *Day 2048*, exactly 2048 days after the programming dual Alice and Bob first met. They fell in love at first sight and have been together since *Day 0*, the day they first met.

Alice and Bob do everything together — watching films, eating out, and, of course, programming. On *special days*, they dine out at a high-class restaurant and afterwards work together on an extra tough programming problem.

The special days include:

- Anniversaries of Day 0, **excluding** Day 0 itself. For example, if "Day 0" is 13th January 2011, then every 13th day of January starting from 2012 would be a special day.
- Day $100n$, where n is any positive integer. For example, if "Day 0" is 13th January 2011, then 23rd April 2011, 1st August 2011, and so on, would be special days.
- Alice's birthdays.
- Bob's birthdays.

Note that if more than one of the special events listed above occur on the same day, the day is still a special day and you should not count it more than once.

Alice has just asked Bob whether he remembers how many special days there are since day 1. Given the date of "Day 0" and the birthdays of Alice and Bob, you are to help Bob answer Alice's question. If Bob fails to answer correctly, Alice is going to get really angry!

Hint: There are 28 days in February in a common year and 29 days in February in a leap year. Year Y is a leap year if and only if Y is a multiple of 400, or Y is a multiple of four and is not a multiple of 100.

Input

The first line of input consists of the date of *Day 0*, given in the format `DD/MM/YYYY`. Day 0 is no earlier than 1st January 1950 and is no later than 13th January 2011.

The second line of input consists of the date (month + day) of Alice's birthday. The third line of input consists of the date (month + day) of Bob's birthday. These birthdays are given in the format `DD/MM`.

Output

Output one single integer, the number of special days between Day 1 and Day 2048 (the current day) inclusive.

Examples

input

```
13/01/2011
23/04
31/12
```

output

```
35
```

input

```
25/01/1999
29/02
25/01
```

output

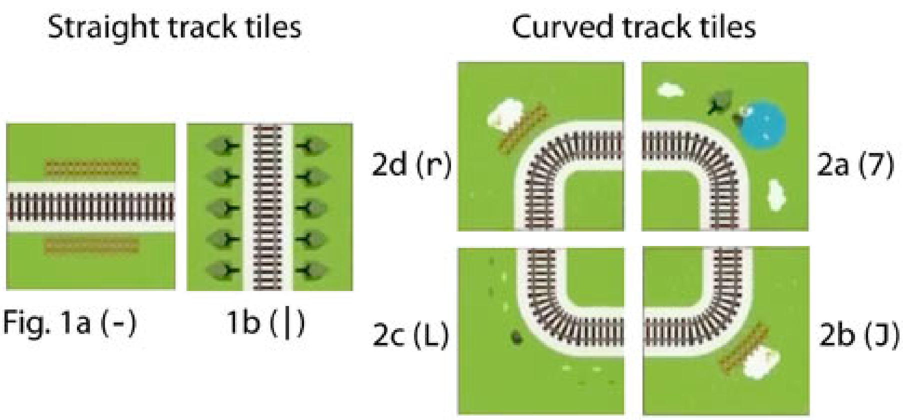
```
26
```

G Monorail

Time Limit: 1 second
Memory Limit: 64 MB
Input: standard input
Output: standard output

Monorail is a game featured in the Korean game show "The Genius". The main objective of Monorail is to form a loop using straight and curved track tiles. Here, we present a modified version of the game.

Percy is the only player and is given N straight tile and M curved track tiles. To win the game, he must place ALL of his tiles in a grid not larger than 100×100. Together, the tracks must be connected to form one single loop.



For example, a possible solution for $N = 2$ and $M = 8$ is shown below:

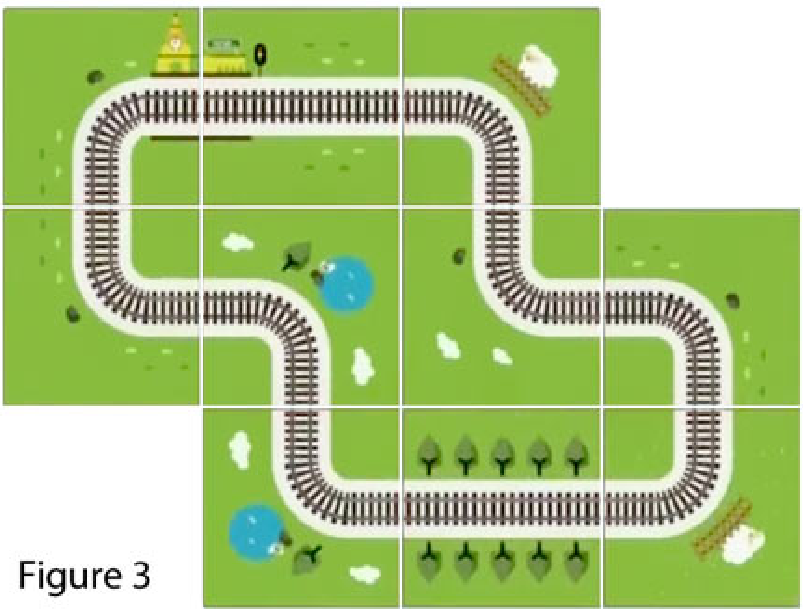


Figure 3

Help Percy determine if a solution exists, and if so, propose a valid solution.

Input

The only line consists of two integers N and M, the number of straight track tiles and curved track tiles respectively.

For all input, $0 \leq N, M \leq 100$ and $N + M > 0$.

Output

If there is no solution, output `Impossible`.

Otherwise, output any valid solution. The first line should contain two integers, H and W, separated by a space. You are free to choose $1 \leq H \leq 100$ and $1 \leq W \leq 100$ as long as the grid is large enough for your solution. It is guaranteed that if there is a solution, then there exists a solution that fits in a 100×100 grid.

Then output the solution grid of height H and width W, consisting of the following characters:

- `.`: Empty cell
- `-`: Straight track tile (Figure 1a)
- `|`: Straight track tile (Figure 1b)
- `7`: Curved track tile (Figure 2a)
- `J`: Curved track tile (Figure 2b)
- `L`: Curved track tile (Figure 2c)
- `r`: Curved track tile (Figure 2d)

Note: `J`, `L` and `r` are case-sensitive.

Examples

input

```
2 8
```

output

```
3 4
r-7.
L7L7
.L-J
```

input

```
0 4
```

output

```
4 4
....
.r7.
.LJ.
....
```

input

```
6 2
```

output

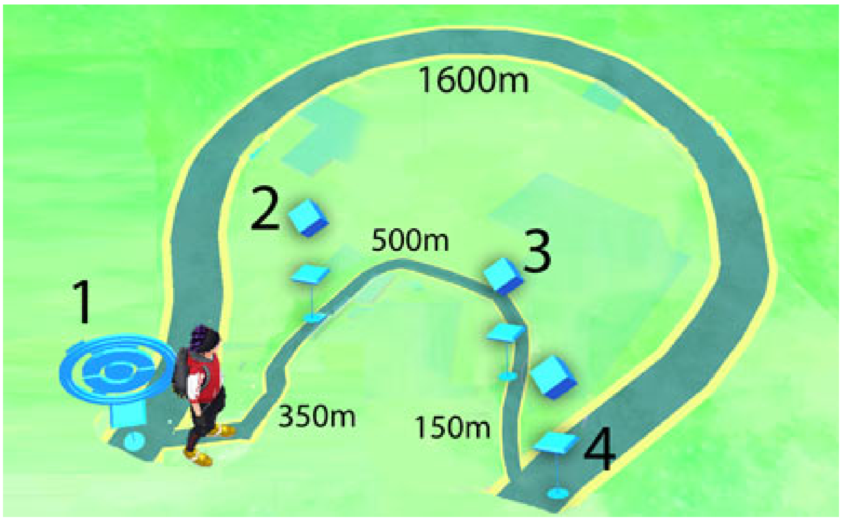
```
Impossible
```

H Pokemon GO

Time Limit: 2 seconds
Memory Limit: 256 MB
Input: standard input
Output: standard output

Be a Pokemon trainer! In Pokemon GO, you can collect eggs from Pokestops. An egg can be hatched by putting it in an incubator, followed by walking a certain distance in the real world.

Being a smart trainer, Ian is trying to find ways to maximize his egg hatching efficiency. The city has N pokestops, numbered 1 to N, and there are M two-way routes connecting pairs of Pokestops. The i^{th} route is L_i meters long. No two routes connect the same pair of Pokestops. Also, no route connect a Pokestop to itself. It is always possible to reach any Pokestop from a Pokestop.



Ian is now at Pokestop A and he wishes to arrive at Pokestop B after the walk. Help him find a path that is exactly K meters long. While he is allowed to visit a Pokestop zero or more times and use a route zero or more times, he cannot return in the middle of a route. There is **no need** to maximize the number of Pokestop visits.

Input

The first line contains three integers N, M, K – the number of Pokestops, the number of routes, and the required path length respectively. ($2 \leq N \leq 1000$, $1 \leq M \leq \min(5000, \frac{N(N-1)}{2})$, $K \in \{2000, 5000, 10000\}$)

The second line contains two integers A and B – the starting and ending Pokestops. A and B are not necessarily different.

The next M lines describe the routes. Each line contains three integers X_i, Y_i, L_i, meaning that the route connects Pokestops X_i and Y_i and its length is L_i. ($1 \leq X_i$, $Y_i \leq N$, $X_i \neq Y_i$, $1 \leq L_i \leq 10^9$)

Output

If there is no path of exactly K meters long, output `Impossible`.

Otherwise output any valid path, which is the order of the Pokestops that Ian should visit.

Examples

input

```
4 4 2000
1 4
1 2 350
2 3 500
3 4 150
1 4 1600
```

output

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

input

```
3 3 5000
2 2
1 2 480
1 3 360
2 3 90
```

output

```
Impossible
```

Note

The first sample corresponds to the image in the statement. The sample output path length = $350+500+500+500+150=2000$.

In the second sample, it is easy to see that there is no path of length 5000 because all route lengths are multiples of 30.

I

RNG

Time Limit: 1 second
Memory Limit: 64 MB
Input: standard input
Output: standard output

RNG stands for **R**andom **N**umber **G**enerator. Most RNGs are, in fact, psuedorandom number generators, which generate numbers that look random but are actually not. One of the simplest RNGs, which is called a linear congruential generator, works as follows:

1. The values of a, b, m, and X_1 are determined.
2. The sequence $\{X_n\}$ is generated using the rule $X_{i+1} = (aX_i + b) \bmod m$.

In this problem, we consider a slightly more complicated RNG, which works as follows:

1. The values of a, b, c, d, m, and X_1 are determined.
2. The sequence $\{X_n\}$ is generated using the rule $X_{i+1} = (aX_i^3 + bX_i^2 + cX_i + d) \bmod m$.

A sequence of 15 integers was generated using the aforementioned RNG. The sequence consisted solely of integers 1, 2, 3, and 4. Unfortunately, some data — including the parameters a, b, c, d, m, and X_1, as well as some of the values of the integer sequence — is lost.

Your task is to recover **any** possible values of the parameters a, b, c, d, m, and X_1, given the remaining values of the integer sequence. You do not need to recover the values of the sequence, since the sequence only depends on the parameters.

Your parameters must satisfy the following constraints:

- $0 \le a, b, c, d \le 1023$
- $1 \le m \le 65535$
- Let the generated sequence be $\{Y_n\}$. Then $1 \le Y_i \le 4$
- If the value of X_i is not lost, then $Y_i = X_i$

Input

The first and only line of input consists of a string with exactly 15 characters. The i-th character is `?` if the value of X_i is lost. Otherwise, the i-th character is one of `1`, `2`, `3`, `4`, representing the value of X_i.

It is guaranteed that the first character, representing the value of X_1, is a `?`.

Output

If there exist valid values for the parameters a, b, c, d, m, and X_1, output their values in that order separated by spaces.

If there is more than one solution, you may output any one.

If there is no solution, output `-1 -1 -1 -1 -1 -1`.

Examples

input	output
`????12341234123`	`6 4 7 5 20 1`
`?222?222?222???`	`0 0 1 0 3 2`
`?23411111111111`	`-1 -1 -1 -1 -1 -1`

Posters

Time Limit: 1 second
Memory Limit: 64 MB
Input: standard input
Output: standard output

Alex and Bob are classmates and they are equally enthusiastic about football.

Needless to say, both of them have a vast collection of posters featuring football stars. They want to stick their posters on the classroom bulletin board, but the class teacher, when in a bad mood, may get annoyed with the posters and demand to have the posters removed.

To "not take any risk", they decide to install a rectangular mini bulletin board in the classroom. The bulletin board is H centimeters long and W centimeters wide. Also, to avoid unwanted attention, they are going to each stick one poster only on the bulletin board.

Both Alex's and Bob's posters are rectangular. Alex's poster is h_a centimeters long and w_a centimeters wide, where Bob's is h_b centimeters long and w_b centimeters wide.

Alex is going to stick his poster first, since it is he who came up with this brilliant idea of installing a mini bulletin board to reduce risk. He looks at his poster, then Bob's, and is suddenly not sure about whether he should let Bob stick his. He formulates the following two plans:

- Plan A. If Bob's poster features football stars that he likes, he will try to stick his poster so that Bob will have space to stick his.
- Plan B. If Bob's poster features football stars that he hates, he will try to stick his poster so that Bob will not have space to stick his.

Note that both Alex and Bob are stubborn people. There are several rules of sticking posters that they follow:

- The posters must lie within the bulletin board. A poster can touch the board margin but must not exceed it by even a nanometer.
- Bob's poster cannot cover Alex's poster, but the posters are allowed to touch.
- The posters cannot be rotated or reflected.

Given the dimensions of the bulletin board and the two posters, determine if Alex can successfully execute his two plans. If he cannot stick his poster on the bulletin board, both of his plans are considered to have failed, *regardless of whether Bob can stick his poster afterwards*.

Input

The first line of input consists of two space-separated integers H and W, the dimensions (in centimeters) of the bulletin board.

The second line of input consists of two space-separated integers h_a and w_a, the dimensions (in centimeters) of Alex's poster.

The third line of input consists of two space-separated integers h_b and w_b, the dimensions (in centimeters) of Bob's poster.

It is guaranteed that $10 \le H, W, h_a, w_a, h_b, w_b \le 150$.

Output

On the first line of output, output `Yes` if Alex can successfully execute Plan A and `No` if otherwise.

On the second line of output, output `Yes` if Alex can successfully execute Plan B and `No` if otherwise.

Examples

input

```
20 30
15 10
10 20
```

output

```
Yes
Yes
```

input

```
20 30
21 10
20 30
```

output

```
No
No
```

input

```
50 100
30 20
30 20
```

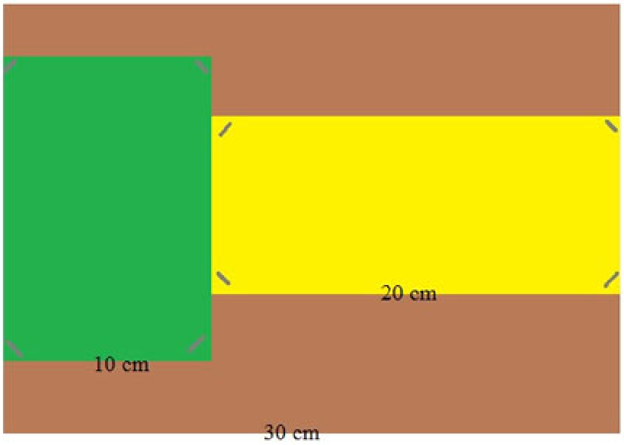
output

```
Yes
No
```

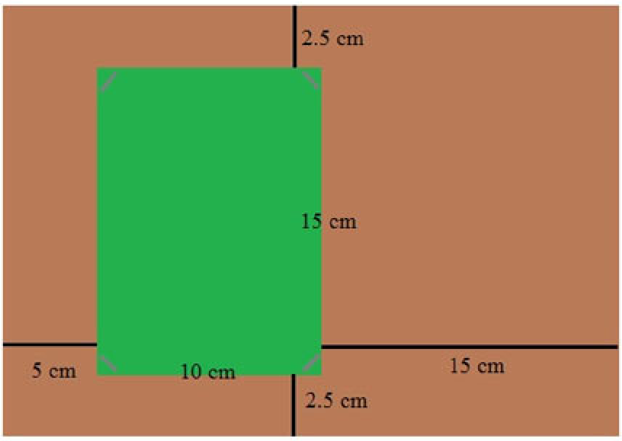
Note

For sample case 1, Alex can execute both plans by placing his poster as shown below.

Plan A



Plan B



Therefore, you should output `Yes` on both lines.

K Lattice Points

Time Limit: 1 second
Memory Limit: 64 MB
Input: standard input
Output: standard output

Jeremy loves geometry and counting. Unsurprisingly, he wants to do both at the same time!

On a Cartesian coordinate plane, Jeremy now draws a circle of radius R, centered at the origin. He wants to know how many lattice points have a distance *no further than* D units from the circumference of the circle he has just drawn. A lattice point on the Cartesian coordinate plane is a point with integer coordinates.

Jeremy can only draw small circles and count the points one by one. When R and D get large, he has no idea what the answer is. Help Jeremy!

Input

The first and only line of input consists of two integers R and D.

For all input, $1 \leq R \leq 500000$, $0 \leq D \leq 500000$.

Output

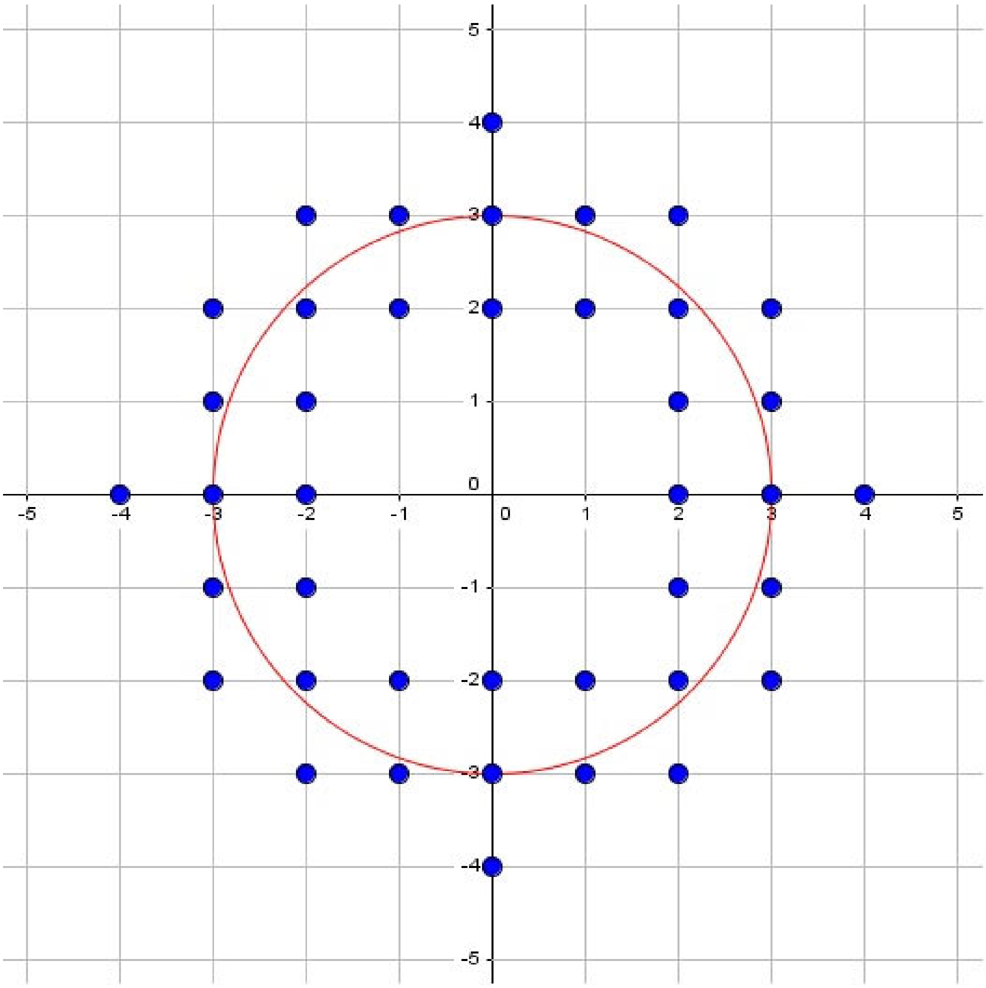
Output one single integer, the number of lattice points no further than D units from the circumference of a circle of radius R and centered at the origin.

Hint: Use 64-bit integers (Pascal: `int64`, C++: `long long`).

Examples

input	output
`3 1`	`40`
`2 4`	`113`
`123456 78901`	`122406524248`

Note



Pictured is sample 1. The small dots indicate the lattice points that should be counted towards the answer.

L Textbook Game

Time Limit: 2 seconds
Memory Limit: 128 MB
Input: standard input
Output: standard output

Marcus and Vincent are having lunch at a restaurant, but none of them wants to pay. They decided to play a game to decide who is going to pay.

They recalled a game in Running Man: using a textbook, each of them flips open a spread (two facing pages) without looking. The faces on the spread is then counted and the person whose page contains more faces wins.

Running Man E252

The textbook that they have chosen has N spreads, numbered 1 to N. The spread numbered i has F_i faces on it. Being a smart student, Vincent would like to know his chance of winning beforehard. Help Vincent by computing the chance of winning, a draw, and losing if Marcus opened spread i, for every $i = 1, 2, ..., N$. Assume that each spread has equal chance of being flipped open by Vincent.

Input

The first line contains an integer N – the number of spreads in the textbook. ($1 \le N \le 10^5$)

The second line contains N integers. The i^{th} integers is F_i – the number of faces on spread i. ($0 \le F_i \le 10^7$)

Output

Output N lines. On the i^{th} line output three numbers separated by spaces – Vincent's probability of a win, a draw, and a loss, if Marcus opened spread i.

Your answer will be accepted if the relative error or absolute error, whichever is less, is not greater than 10^{-6} for all $3N$ numbers.

Examples

input

```
4
1 1 3 0
```

output

```
0.250000 0.500000 0.250000
0.250000 0.500000 0.250000
0.000000 0.250000 0.750000
0.750000 0.250000 0.000000
```

input

```
3
2 2 2
```

output

```
0 1 0
0 1 0
0 1 0
```

Note

In the first sample,

- If Marcus flipped open one the first two spreads (1 face), Vincent has 25% chance of winning (spread 3), 50% chance of a draw (spreads 1 and 2), and 25% chance of losing (spread 4).
- If Marcus flipped open spread 3 (3 faces), Vincent has 25% chance of a draw (spreads 3), and 75% chance of losing (spreads 1, 2, 4).
- If Marcus flipped open spread 4 (no faces), Vincent has 75% chance of winning (spreads 1, 2, 3) and 25% chance of a draw (spread 4).

In the second sample, unfortunately all spreads have equal number of faces. Marcus and Vincent will never be able to decide who is going to pay.