NTU

Competitive Programming Club

臺灣大學程式解題社

National Taiwan University Competitive Programming Club

NTUCPC Speed Run! #2

November 9, 2025

Problem	Problem Name
A	Water Rocket
B	Election
C	Secret Recipe of the Great Elixir
D	Party
E	The Wizard’s Polynomial Challenge
F	Misleading Ad
G	Master of Imitation
H	Running Laps
I	Tic-Tac-Toe
J	Queue at the Restaurant
K	Git Graph
L	Center-pivot Irrigation
M	Solar System
N	Sniper Showdown
O	Stock Tycoon’s Profit Tracker
P	System of Equations under Modulo
Q	Train Transportation System
R	Secret Theft
S	Time Split
T	Delivery Drone Range Optimization
U	Portals
V	The God of Mathematics
W	Fruit Kart
X	Forest of Equilibrium
Y	Painter
Z	Cutscenes

This page is intentionally left blank.

A. Water Rocket

Problem ID: rocket

You are a member of the Intergalactic Water Rocket Rescue Team. During a test mission on Planet Aqua, your team launched a water rocket vertically into the sky.

Unfortunately, due to a sensor glitch, you only know that the rocket reached a maximum height of h meters before starting to fall back down. The gravity on Planet Aqua is similar to Earth's, causing a downward acceleration of 9.8 m/s^2. This means the rocket's upward velocity decreases by 9.8 meters per second every second until it reaches its peak.

To calibrate the launch system for future missions, you must determine the rocket's initial launch velocity — the speed it had the moment it left the ground.

Input

The input only contains one line with one integer h, which is the maximum height reached by the rocket in meters.

- $0 < h \le 10^9$

Output

Output a single number: the initial velocity of the rocket in meters per second.

Your answer will be considered correct if its absolute or relative error does not exceed 10^{-6}.

Sample Input 1	Sample Output 1
49	30.99032106965

This page is intentionally left blank.

B. Election

Problem ID: election

The annual fruit kingdom representative election is about to kick off in a few days. However, as the population of the kingdom grows exponentially, counting the votes manuall in real life won't be feasible this year.

There are at most 26 candidates and each candidate is represented by a distinct uppercase Latin alphabet. The election committee of the kingdom plans to have a new automatic tool that supports the following updates:

- Add a vote for a certain candidate.
- Report all the candidates with the most number of votes in the alphabetic order.

Can you complete the task for the kingdom?

Input

The first line of the input contains one integer n ($2 \leq n \leq 10^5$) denoting the number of updates. Next, n lines follow. Each of the lines represents an update in one of the following two formats:

- `+` C: add a vote for candidate C. There is a space between the plus character and the alphabet of the candidate.
- `?`: query the candidates with the most number of votes. This type of update will not be the first update and there is at least one update of this type.

Output

For each query, output a string on a separate line with all the candidates with the most number of votes in alphabetic order. There should be no space between the characters in each query.

Sample Input 1

```
9
+ W
+ M
+ K
?
+ W
+ W
+ M
+ K
?
```

Sample Output 1

```
KMW
W
```

Sample Input 2

```
8
+ A
+ B
?
+ A
?
+ B
+ B
?
```

Sample Output 2

```
AB
A
B
```

Sample Input 3

```
6
+ F
+ R
+ U
+ I
+ T
?
```

Sample Output 3

```
FIRTU
```

C. Secret Recipe of the Great Elixir

Problem ID: elixir

The Great Elixir is the best-selling product at the famous Crabby Diner, run by Mr. Crabbo.

You are a secret agent working for the rival restaurant, Chum Cavern, founded by the notorious Dr. Planko. After years of plotting, you've finally infiltrated the secret lab beneath the Crabby Diner and stolen two mysterious solution samples that are used to create the Great Elixir. You've also discovered that the Great Elixir is obtained by mixing the two solutions in a specific ratio.

The first solution contains $a\%$ magical essence, and the second contains $b\%$ magical essence.

After careful analysis, you've determined that the final Elixir has $c\%$ magical essence when the two solutions are mixed in the correct proportion.

Dr. Planko now needs your help to determine exactly how many grams of each original solution must be used to create exactly k grams of the Great Elixir that contains $c\%$ magical essence.

The concentration of magical essence is defined as:

$$\frac{\text{mass of magical essence}}{\text{total mass of solution}}$$

Mixture rules:

- The total mass of the mixture is equal to the sum of the masses of the individual components.
- The total magical essence is equal to the sum of the magical essences from the two solutions.

It is guaranteed that exactly one valid solution exists.

Input

A single line containing four space-separated integers a, b, c, and k:

- a is the percentage of magical essence in the first solution.
- b is the percentage of magical essence in the second solution.
- c is the desired percentage of magical essence in the final mixture.
- k is the total mass of the Great Elixir to be produced in grams.

- $0 \le a, b, c \le 100$
- $a \ne b$
- $1 \le k \le 10^9$

Output

Output a single line containing two space-separated non-negative numbers x and y, where x is the mass of the first solution and y is the mass of the second solution, both in grams.

It is guaranteed that there's exactly one valid solution.

Your answer will be considered correct if its absolute or relative error does not exceed 10^{-6}.

Sample Input 1	Sample Output 1
`25 75 35 100`	`80.0000000000000000 20.0000000000000000`

D. Party

Problem ID: party

NTU City is a city passionate about sports, and every citizen is an outstanding athlete. As a tradition, NTU City holds a city-wide sports competition every year to determine who the strongest athlete in the city is. To celebrate, the overall champion of the competition will host a party and invite all of their friends to attend. Additionally, the champion allows each of their friends to invite any number of their own friends to the party.

As a citizen of NTU City, Xiao Li is curious: assuming only the party host and the people who are invited are allowed to attend the party, if we know who is friends with whom in the city, and who the champion is, what is the maximum number of people who could show up at the party?

Input

The first line contains three positive integers N, M, and W, representing the total number of citizens in NTU City, the total number of friendship pairs in the city, and that the W-th citizen is the champion of the competition.

The next M lines each contain two positive integers x_i and y_i, indicating that the x_i-th citizen and y_i-th citizen are friends.

- $1 \leq N \leq 2 \times 10^5$
- $0 \leq M \leq 2 \times 10^5$
- $1 \leq x_i, y_i, W \leq N$
- $x_i \neq y_i$
- $\forall 1 \leq i, j \leq M, i \neq j, (x_i, y_i) \neq (x_j, y_j)$ and $(x_i, y_i) \neq (y_j, x_j)$

Output

Output a single positive integer, indicating the maximum number of people who could possibly appear at the party.

Sample Input 1

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

Sample Output 1

```
4
```

Sample Input 2

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

Sample Output 2

```
5
```

E. The Wizard's Polynomial Challenge

Problem ID: poly

In the magical kingdom of Algoria, the wise wizard PolyNim has conjured a powerful polynomial spell defined as:

$$P(x) = a_0 + a_1x + a_2x^2 + \cdots + a_nx^n$$

To test your mastery of polynomial magic, he will ask you multiple questions. In each question, you are given an integer k, and you need to compute the value of the polynomial $P(x)$ at $x = k$. Since the result may be very large, you only need to report the last 9 digits of the result. Moreover, you do not need to output any leading zeros. For example, if the result is $1,000,000,001$, you should output 1. And if the result is $1,000,000,000$, you should output 0.

Input

The first line of the input contains an integer n, the degree of the polynomial. The second line contains $n+1$ of integers $a_0, a_1, \cdots, a_n$, the coefficients of the polynomial. The third line contains an integer Q, the number of queries. The next Q lines each contains an integer k, the point at which to evaluate the polynomial and its derivative.

- $0 \leq n \leq 2 \cdot 10^5$
- $0 \leq a_i \leq 10^9$ for all $0 \leq i \leq n$
- $a_n \neq 0$
- $1 \leq Q \leq 2 \cdot 10^5$
- $0 \leq k \leq 10^9$ for all queries
- $n \cdot Q \leq 2 \cdot 10^5$

Output

For each query, output a line containing a single integer representing the last nine digits of $P(k)$ without leading zeros.

Sample Input 1

```
2
1 2 3
3
2
0
10
```

Sample Output 1

```
17
1
321
```

Sample Input 2

```
0
5
5
20
40
27
33
99
```

Sample Output 2

```
5
5
5
5
5
```

Sample Input 3

```
7
7 0 2 90 54 66 38 20
4
0
16
20
6
```

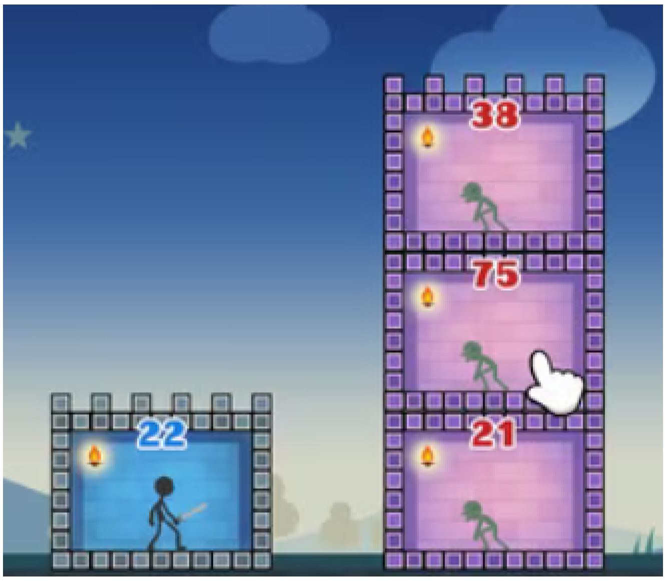
Sample Output 3

```
7
79357447
252560807
7974367
```

F. Misleading Ad

Problem ID: ad

Xiao Ming loves playing mobile games. However, he recently noticed that some games use misleading advertisements that are completely unrelated to the actual gameplay, just to attract downloads. Feeling deceived and frustrated, Xiao Ming decided to turn these fake ads into programming problems!



One of the advertisements Xiao Ming saw is shown in the image above. It features a kind of arithmetic tower-conquering game. In the game, there are N towers arranged from left to right. On the i-th tower, there are k_i enemies and s_i swords. Each enemy has a combat power of $a_{i,j}$.

The player starts with an initial combat power of X and begins from the leftmost tower. During gameplay, the player can choose to fight any enemy on the current tower who hasn't been defeated yet. If the player's current combat power is **not less than** the enemy's, the player defeats the enemy and the player's combat power increases by the enemy's combat power. Otherwise, the player loses and the game ends.

Besides fighting enemies, the player may also pick up an uncollected sword from the current tower at any time, as long as at least one sword remains uncollected on that tower. Doing so **doubles the player's current combat power**.

Once the player has defeated all enemies on the current tower, they may proceed to the next tower. However, all uncollected swords on a tower may only be picked up before moving to the next tower, and players cannot go back to previous towers to pick up any uncollected swords.

Now, your task is to help determine: What is the maximum number of towers the player can complete, assuming they play with the best possible strategy?

Input

The first line of input contains two positive integers N and X, representing the number of towers and the player's initial combat power.

The next N lines describe each tower. With the i-th line starting with a positive integer k_i and a non-negative integer s_i, indicating the number of enemies and swords on the tower. Follows by k_i positive integers: $a_{i,1}, a_{i,2}, \dots, a_{i,k_i}$, which are the combat powers of the enemies.

- $1 \leq N \leq 10^5$
- $1 \leq \sum_i (k_i + s_i) \leq 10^6$
- $k_i \geq 1, s_i \geq 0$
- $1 \leq a_{i,j}, X \leq 10^9$

Output

Output a single non-negative integer indicating the maximum number of towers the player can completely clear.

Sample Input 1	Sample Output 1
1 22 3 0 21 75 38	1

Sample Input 2	Sample Output 2

3 10
2 0 5 12
3 1 48 25 99
2 0 300 500
	3

Sample Input 3

```
3 10
2 0 5 12
3 1 48 25 99
2 0 300 1000
```

Sample Output 3

```
2
```

This page is intentionally left blank.

G. Master of Imitation

Problem ID: imitator

Alice is a renowned master of imitation. She's skilled at mimicking others with remarkable accuracy in all aspects. Her impressive imitation skills come from her constant practice, and as such, she puts great emphasis on honing her craft.

One day, Alice sets herself a new practice challenge. She notices that Bob has a sequence of length N, where each number from 1 to N appears exactly once. Alice decides to imitate Bob's sequence, so she constructs her own sequence of length N, also containing the numbers from 1 to N exactly once. However, she quickly realizes that her sequence looks quite different from Bob's.

To live up to her title as the master of imitation, Alice decides to transform her sequence into Bob's by performing a series of adjacent element swaps. For example, she can swap the adjacent elements $(3, 2)$ in $\{1, 3, 2, 4, 5\}$ to get $\{1, 2, 3, 4, 5\}$.

Can you help Alice calculate the minimum number of adjacent swaps she needs to perform to make her sequence identical to Bob's?

Input

The first line contains a positive integer N.

The second line contains N positive integers $a_1, a_2, ..., a_N$, representing Alice's sequence.

The third line contains N positive integers $b_1, b_2, ..., b_N$, representing Bob's sequence.

- $1 \leq N \leq 5000$
- $1 \leq a_i, b_i \leq N$
- $a_i \neq a_j, b_i \neq b_j, \forall i \neq j$

Output

Output a single integer: the minimum number of adjacent swaps Alice needs to transform her sequence into Bob's.

Sample Input 1	Sample Output 1

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

H. Running Laps

Problem ID: laps

In the sports ground of NTU, there is a circular track of length L meters. Alice and Bob are two friends who are running on the track.

Figure: A satellite image of the track of NTU. Source: Google Earth.

They are both running clockwise, Alice is running v_A meters per second and Bob is running v_B meters per second. In the meantime, Alice is located at x_A meters from the starting point and Bob is located at x_B meters from the starting point. Even though they are good friends, neither of them is willing to change their speed. However, they still want know where they will meet for the first time and they are asking you to calculate that for them! As you can see, they are asking you to calculate a pretty easy problem so they are probably not good at math and don't understand floating point numbers or fractions. If the location they meet is not an integer in meters from the starting point, you should report the first integer meters from the starting point that they will run to after they meet.

Input

Input only consists of one line consisting of five integers: L, v_A, v_B, x_A, and x_B.

- $1 \leq L \leq 10^9$ (Yes, the track can be very long.)
- $1 \leq v_A, v_B \leq 10^9$ (And also yes, they can run very fast.)
- $0 \leq x_A, x_B < L$
- $x_A \neq x_B$

Output

If Alice and Bob will never meet, output `:(`. Otherwise, output an integer being the location where they will meet for the first time or the first integer meters from the starting point that they will run to after they meet.

Sample Input 1

```
10 4 2 3 1
```

Sample Output 1

```
9
```

Sample Input 2

```
10 4 2 0 1
```

Sample Output 2

```
2
```

Sample Input 3

```
10 4 2 0 2
```

Sample Output 3

```
4
```

Sample Input 4

```
10 5 1 0 1
```

Sample Output 4

```
2
```

Sample Input 5

```
10 1000000000 1000000000 0 9
```

Sample Output 5

```
:(
```

I. Tic-Tac-Toe

Problem ID: tictactoe

Tic-Tac-Toe, the game that everyone knows and loves. However, in this problem, we are not playing the standard game. Instead, we are having a massive board!

(1,1)	(1,2)	(1,3)	$\cdots$	$(1,N)$
(2,1)	(2,2)	(2,3)	$\cdots$	$(2,N)$
(3,1)	(3,2)	(3,3)	$\cdots$	$(3,N)$
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$
$(N,1)$	$(N,2)$	$(N,3)$	$\cdots$	(N,N)

In contrast to the regular game, we have a board of size $N \times N$. Then, two players take turns to place their marks on the board like the regular game, the first player goes with `X` and the second player goes with `O`. Another difference from the regular game is that the players can place their marks on a tile that is already occupied. When a tile is marked again, the original mark will be overwritten by the new mark. After each move, the point of each player at that time is the number of horizontal, vertical, and diagonal lines that are formed entirely by the player's marks.

Now, you are given the coordinates of each of the moves made by the players. Your task is to calculate the points of each player after each move.

Input

First line of input consists of two integers N, Q.

Next Q lines each consist of two integers x_i, y_i. (x_i, y_i) indicates the coordinate that the player mark on the overall i-th move. If i is odd, the move is made by the first player, otherwise it is made by the second player.

- $2 \le N \le 1000$
- $1 \le Q \le 10^6$
- $1 \le x_i, y_i \le N$

Output

Output Q lines, each line consists of two integers separated by a space. In the i-th line, the first integer is the points of the first player right after the i-th move, and the second integer is the points of the second player right after the i-th move.

Sample Input 1

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

Sample Output 1

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

Sample Input 2

```
3 8
1 1
1 3
3 3
3 1
2 2
2 2
2 2
2 2
```

Sample Output 2

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

J. Queue at the Restaurant

Problem ID: restaurant

Taiwan is famous for its delicious food. However, even with so many delicacies around, Taiwanese people are also famous for loving to wait at a long queue for food.

In this famous beef noodle restaurant, the kitchen can only cook one customer's order at a time. The i-th customer arrived at time t_i and their order takes w_i minutes to cook. When each customer arrives, if the kitchen is not busy, their order will start being cooked immediately. Otherwise, they will wait at the back of the queue. Everyone follows the queue order and no one cuts in line. The kitchen can start cooking the next customer's order the moment the previous order is finished. Now, for each customer, please calculate the total time they have to wait before the order is finished.

Input

The first line of input consists of an integer N.

The next N lines each consists of two integers t_i, w_i.

- $1 \leq N \leq 10^5$
- $1 \leq t_i, w_i \leq 10^9$
- t is strictly increasing. In other words, $\forall 1 \leq i < N, t_i < t_{i+1}$ 。

Output

Output N lines, each line consists of one integer. The i-th line is the total time that the i-th customer has to wait before their order is finished.

Sample Input 1

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

Sample Output 1

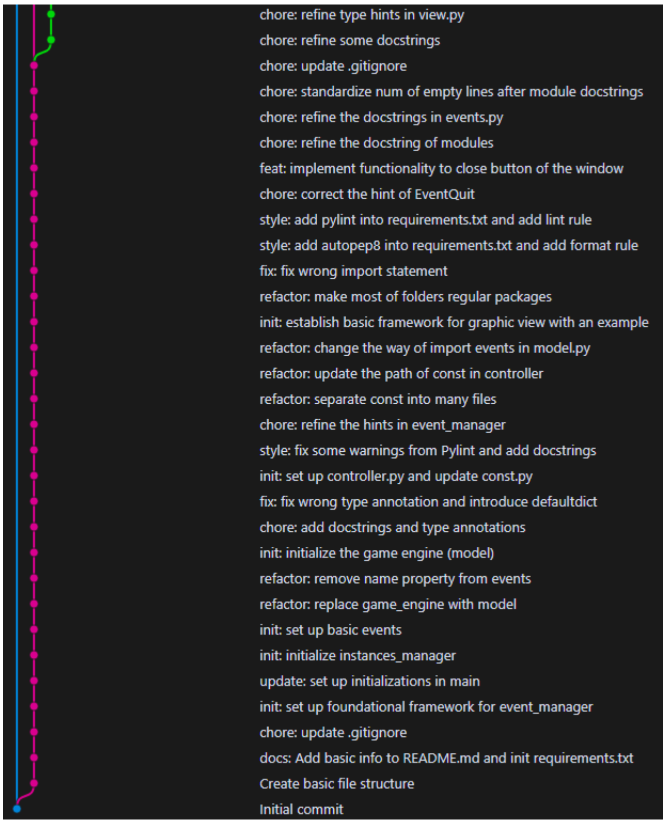
```
5
5
5
```

This page is intentionally left blank.

K. Git Graph

Problem ID: graph

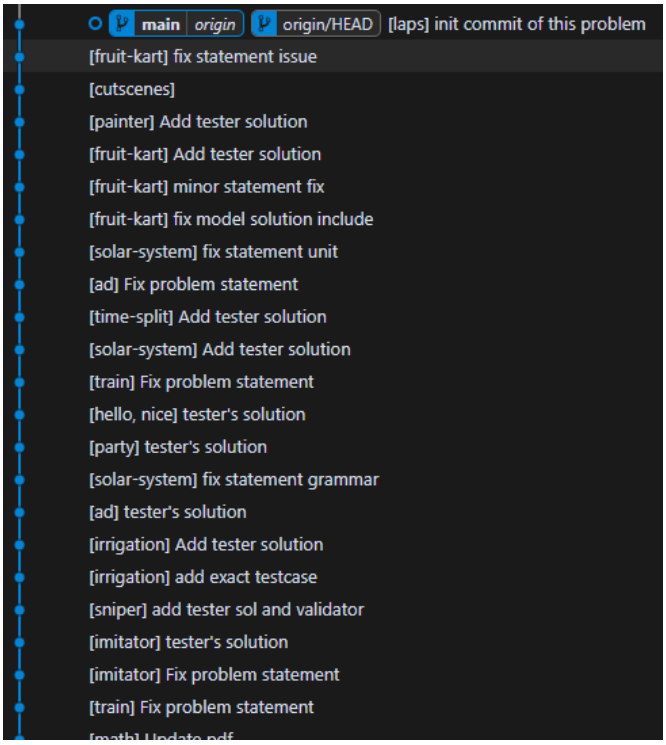
Git is a modern version control system that is used widely in the software development industry. A key feature of Git is the ability to create branches, which are separate lines of development that can be merged back together. This can simplify the development process such that parallel development of features can be done without interfering with each other.



```
chore: refine type hints in view.py
chore: refine some docstrings
chore: update .gitignore
chore: standardize num of empty lines after module docstrings
chore: refine the docstrings in events.py
chore: refine the docstring of modules
feat: implement functionality to close button of the window
chore: correct the hint of EventQuit
style: add pylint into requirements.txt and add lint rule
style: add autopep8 into requirements.txt and add format rule
fix: fix wrong import statement
refactor: make most of folders regular packages
init: establish basic framework for graphic view with an example
refactor: change the way of import events in model.py
refactor: update the path of const in controller
refactor: separate const into many files
chore: refine the hints in event_manager
style: fix some warnings from Pylint and add docstrings
init: set up controller.py and update const.py
fix: fix wrong type annotation and introduce defaultdict
chore: add docstrings and type annotations
init: initialize the game engine (model)
refactor: remove name property from events
refactor: replace game_engine with model
init: set up basic events
init: initialize instances_manager
update: set up initializations in main
init: set up foundational framework for event_manager
chore: update .gitignore
docs: Add basic info to README.md and init requirements.txt
Create basic file structure
Initial commit
```

Figure: A Git graph of the first few commits of a repository.

In a Git repository, to edit the codebase, developers create commits to record the changes they made. A commit is a snapshot of the codebase at a certain point in time. More importantly, all commits except the initial commit will also record the id of the parent commit. This forms a chain if the repository's history is simple and linear.



```
[laps] init commit of this problem
[fruit-kart] fix statement issue
[cutscenes]
[painter] Add tester solution
[fruit-kart] Add tester solution
[fruit-kart] minor statement fix
[fruit-kart] fix model solution include
[solar-system] fix statement unit
[ad] Fix problem statement
[time-split] Add tester solution
[solar-system] Add tester solution
[train] Fix problem statement
[hello, nice] tester's solution
[party] tester's solution
[solar-system] fix statement grammar
[ad] tester's solution
[irrigation] Add tester solution
[irrigation] add exact testcase
[sniper] add tester sol and validator
[imitator] tester's solution
[imitator] Fix problem statement
[train] Fix problem statement
[math] Update pdf
```

Figure: A simple Git graph of this repository. It is linear because no branch is created.

However, in a more complex repository, different developers working on different features may edit the same files. In this case, they should create branches to avoid their edits interfering with each other. When a new commit is created, its parent commit can be set to a commit already with an *offspring*. This will create the branch automatically. The branching will result in one commit being the parent of multiple commits.

In a repository with branching, developers should only deal with the conflict in one go when they merge the branches back together. This creates a special type of commit called a merge commit. A merge commit will have two parents.

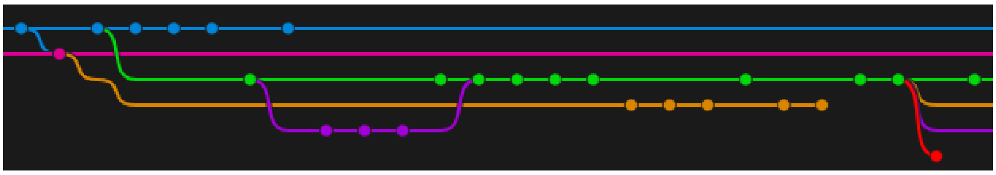


Figure: A more complex Git graph of another repository. It is not linear because branches are created and merged back together.

As you can see, to understand the history of a repository, drawing the graph is a good way to visualize the history. To separate the history of different branches, we can color the nodes of different branches with different colors. We can say that a branch is *dead* if it has been merged into another branch. Also, as you can see in the lower right corner (the red line) of the graph, if a commit does not have other commits whose parent is it, then the branch can also be considered *dead* after the commit.

Throughout a branch's lifetime, it should be assigned a color that is different from all the colors of alive branches. After the branch *died*, the color can be reassigned to other new branches that are created later than the time of death.

Now, we have the history of a Git repository. However, we only have a list of the commits. For each commit, you are given the timestamp and the index of the parent commit. Your task is to determine how many colors are needed to color the graph following the stated rules.

Input

The first line of input consists of an integer N representing the number of commits in the repository.

The second line consists of an integer T_1 representing the timestamp of the initial commit.

The next $N - 1$ lines of input each represent a commit. Each line begins with a character $type_i$. If $type_i$ is `C`, it will be followed by two integers T_i and P_i. T_i represents the timestamp of the i-th commit and P_i represents that the parent commit is the P_i-th commit.

If $type_i$ is `M`, it will be followed by three integers T_i P_i P'_i. T_i represents the timestamp of the i-th commit and P_i represents that the parent commit is the P_i-th commit. P'_i represents that the other parent commit is the P'_i-th commit.

- $1 \le N \le 10^6$
- $0 \le T_i \le 10^9$
- It is guaranteed that all T_i are distinct.
- $1 \le P_i, P'_i \le N$
- $P_i \ne i$
- $P'_i \ne i$
- $P_i \ne P'_i$
- $T_{P_i} < T_i$
- $T_{P'_i} < T_i$

Output

The output consists of one integer consisting of the minimum number of distinct colors in order to color the graph.

Sample Input 1

```
3
1
C 2 1
M 3 1 2
```

Sample Output 1

```
2
```

Sample Input 2

```
5
1
C 10 1
C 9 1
C 8 1
C 7 1
```

Sample Output 2

```
4
```

Sample Input 3

```
3
1
M 3 1 3
C 2 1
```

Sample Output 3

```
2
```

L. Center-pivot Irrigation

Problem ID: irrigation

When it comes to the irrigation infrastructure for agriculture, center-pivot irrigation is an effective alternative to traditional canals. This method utilizes sprinklers in the center to water nearby fields, creating large green circles when viewed from above, called crop circles.

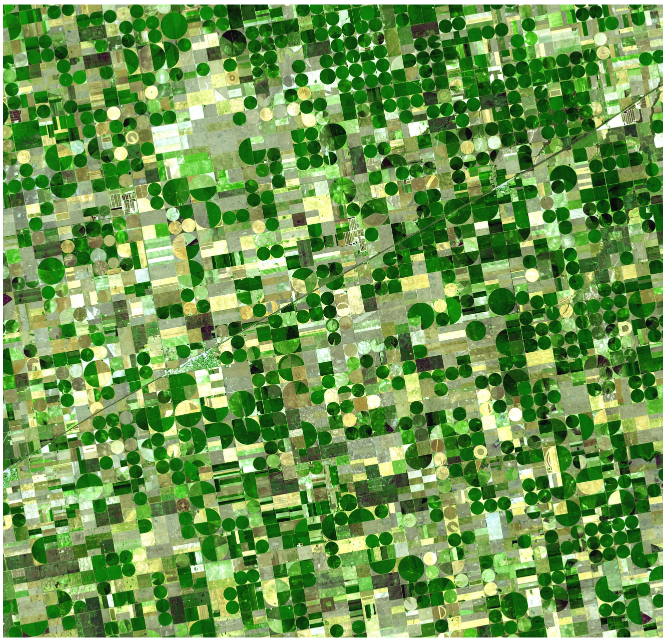


Figure: A satellite image of circular fields characteristic of center pivot irrigation in Kansas, United States.

The fruit kingdom owned n crop circles located around the field. The i-th of the circle is centered at the coordinate (x_i, y_i) and with radius r_i. These crop circles might intersect, or even contain one another.

However, the king of the kingdom, the Legendary Mango King, would like to build a railway on a straight line passing (s_x, s_y) and (t_x, t_y) to improve their transportation system. Although the exact segment is not determined yet, the King lest he damage the traditional agricultural development of the kingdom, so the King would like to know how many crop circles intersect with the path of the straight line.

As you are known as the most prestigious scholar in the realm, the Legendary Mango King asks for your hand to complete this task.

Input

The first line of the input contains five integers, n, s_x, s_y, t_x, t_y. Each of the following n lines contains three integers, x_i, y_i, r_i, on the i-th line. The definition of the variables follows the previous statments.

- $1 \leq n \leq 10^5$
- $-10^4 \leq s_x, s_y, t_x, t_y, x_i, y_i \leq 10^4$
- $1 \leq r_i \leq 10^4$
- $(s_x, s_y) \neq (t_x, t_y)$

Output

Output a single non-negative integer indicating the number of crop circles intersecting with the straight line. Even if multiple circles are of the same coordinate and radius, they should still be counted multiple times.

Sample Input 1

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

Sample Output 1

```
2
```

Sample Input 2

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

Sample Output 2

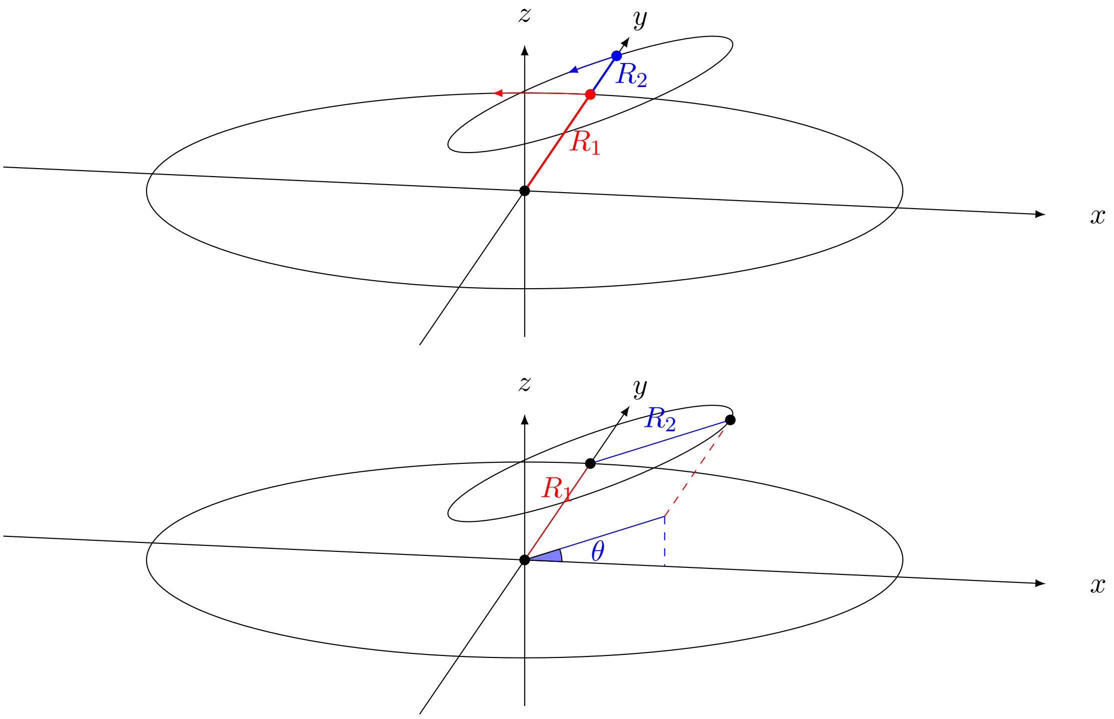
```
3
```

M. Solar System

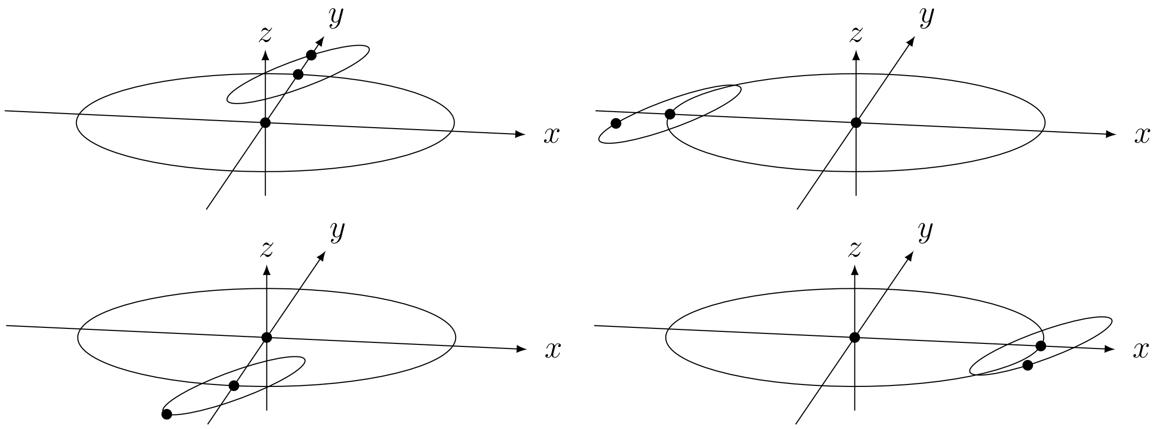
Problem ID: solar-system

In our solar system, the Sun, the Earth, and the Moon, form an intricate system which has shaped our idea of days, months, and years throughout the centuries. This type of planetary system is quite typical in the Milky Way. Planes orbit the stars and satellites (moons) orbit planets. And when the stars align... No, when the star, the planet, and the satellite align, an eclipse happens.

Mathematically speaking, we can approximate the system using five parameters: the two radii of the orbits, the two orbital periods of the bodies, and the angle between the two planes of orbits. Consider a fixed 3D coordinate where the star is the origin. The planet moves along the the circle of radius R_1 on the x-y plane and the satellite follows a circle of radius R_2 centered at the planet. However, the two orbits are not necessarily in the same plane; instead, we use θ to represent the angle between two orbital planes. When the planet is along the $+y$ direction of the star, the satellite's orbit passes through the three points $(0, R_1 - R_2, 0)$, $(0, R_1 + R_2, 0)$, and $(-R_2 \cos\theta, R_1, -R_2 \sin\theta)$, as shown in the following figure. The planet and the satellite both revolve counterclockwise at a constant speed when viewed from above (when viewed along the $-z$-axis).



The orbital relationship between the planet and the satellite is fixed; that is to say, the angle between two orbital planes does not change, as shown in the following figures.



Suppose the orbital period of the planet is T_1 Earth days and that of the satellite is T_2 Earth days. Counting from the configuration where they are aligned along the $+y$ axis in the order of: the star, the planet, and the satellite, what would be the angle between them after t Earth days? Report the angle with the planet as its vertex.

Input

The input contains one line with six space-separated integers, $R_1, R_2, T_1, T_2, \theta, t$. The angle between the orbital planes θ is specified in degrees.

- $1 \le R_1, R_2 \le 2 \cdot 10^8$
- $1 \le T_1, T_2 \le 1000$
- $R_1 \ge 2R_2$
- $0 \le \theta < 90$
- $0 \le t \le 10^9$

Output

Output a single real number representing the angle between the star, the planet, and the satellite, after t Earth days. The answer should be in degrees and must fall within the range of $[0, 180]$.

Your answer will be considered correct if the relative or the absolute error of the value does not exceed 10^{-4}. Formally, if your answer is a and the real answer is b, it will be accepted if and only if $\frac{|a-b|}{\max(b,1)} \leq 10^{-4}$.

Sample Input 1

```
20 10 36 8 45 27
```

Sample Output 1

```
60
```

Sample Input 2

```
149597871 384399 365 27 5 1000
```

Sample Output 2

```
73.02050669239190587433
```

Sample Input 3

```
149597871 384399 365 27 5 0
```

Sample Output 3

```
180
```

This page is intentionally left blank.

N. Sniper Showdown

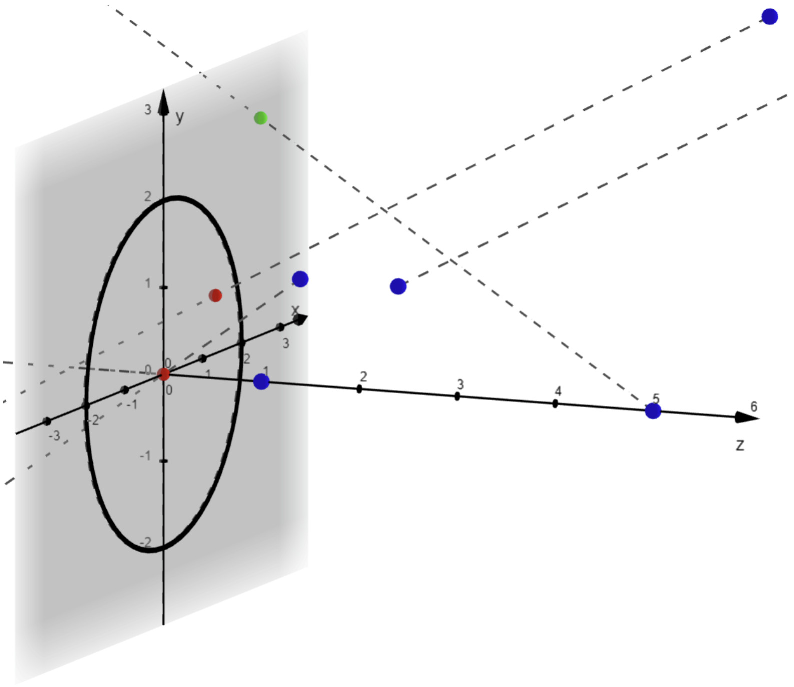
Problem ID: sniper

In the virtual game, Polygon Warfare, Q elite snipers are training in a shooting range. At the center of this arena lies a flat circular target (a bullseye) at the plane $z = 0$. The target is centered at the origin $(0, 0, 0)$ and has a radius of r.

You are given the information about the Q snipers' shots. Each sniper is positioned at some point in front of the target ($z > 0$) and fires a single bullet in a fixed direction. As they are in a virtual game, it is possible that multiple snipers are at the same position. Each bullet travels along an infinite ray defined by its origin and a direction vector.

Your task is to determine if each bullet hits the target, i.e., intersect the circular bullseye on the $z = 0$ plane.

The picture below illustrates 5 bullets fired by 5 snipers, where the blue points represent the starting points of the bullets, and the dot lines represent the trajectories of the bullets.



Input

The first line of the input contains two space-separated integers r and Q, denoting the radius of the circular target and the number of snipers.

Each of the next Q lines contains six space-separated integers: x, y, z, dx, dy, dz —the starting point (x, y, z) of the bullet, and its direction vector (dx, dy, dz).

- $1 \leq r \leq 10^9$
- $1 \leq Q \leq 10^5$
- $z > 0$.
- $-10^9 \leq x, y, z, dx, dy, dz \leq 10^9$.
- It is guaranteed that all of the direction vectors (dx, dy, dz) are not the zero vector.
- It is guaranteed that no bullet's trajectory intersects the $z = 0$ plane at a point whose distance from the origin lies within the range $[(1 - 10^{-6})r, (1 + 10^{-6})r]$.

Output

For each query, output a string on a single line containing ”yes” (without quotation marks) if the bullet hits the target, or ”no” (without quotation marks) if it does not.

Sample Input 1

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

Sample Output 1

```
yes
no
yes
no
yes
```

O. Stock Tycoon’s Profit Tracker

Problem ID: stock

Meet Mr. Goldstein, a legendary stock tycoon known for his sharp trades. Every day, he carefully logs his profit or loss from trading stocks.

Now, Mr. Goldstein wants to calculate his total profit over specific date ranges.

You are given his profit records and a list of queries. Each query asks for the total profit from day l to day r (inclusive). Can you help him compute them efficiently?

Input

The first line contains an integer n, the number of time points (days). The second line contains n space-separated integers: $p_1, \ldots, p_n$, where p_i is the profit or loss on day i. The third line contains an integer q, the number of queries. The next q lines each contain two integers l and r, meaning Mr. Goldstein wants to know the total profit from day l to day r.

- $1 \leq n \leq 10^5$
- $|p_i| \leq 10^9$ for all $1 \leq i \leq n$
- $1 \leq q \leq 10^5$
- $1 \leq l \leq r \leq n$ for all queries

Output

For each query, output a single integer on a new line, the sum of profits from day l to day r, inclusive.

Sample Input 1

```
5
2 5 18 -7 6
5
1 5
2 4
3 3
1 4
4 5
```

Sample Output 1

```
24
16
18
18
-1
```

P. System of Equations under Modulo

Problem ID: modulo

In elementary mathematics, we have learned about system of equations of real numbers. Assuming a system like this:

$$\begin{cases} A_{1,1}x_1 + A_{1,2}x_2 + \cdots + A_{1,n}x_n & = & b_1 \\ A_{2,1}x_1 + A_{2,2}x_2 + \cdots + A_{2,n}x_n & = & b_2 \\ \vdots & = & \vdots \\ A_{n,1}x_1 + A_{n,2}x_2 + \cdots + A_{n,n}x_n & = & b_n \end{cases}$$

This famous problem can be solved with the Gaussian Elimination. The concept of the method is fairly simple. We apply elementary row operations, adding a scalar multiple of a row to another row, on the augmented matrix $[A|b]$. It can be proven that after the operation, the solution to $Ax = b$ will not be affected. Then, to eliminate the n-th term, we look at equations 2 to n, for the i-th equation, we find a coefficient $c = -\frac{A_{i,n}}{A_{1,n}}$ and let the new i-th row be:

$$(cA_{1,1} + A_{i,1})x_1 + (cA_{1,2} + A_{i,2})x_2 + \ldots (cA_{1,n-1} + A_{i,n-1})x_{n-1} + (cA_{1,n} + A_{i,n})x_n = cb_1 + b_i$$

As $cA_{1,n} = -A_{i,n}$, it is actually:

$$(cA_{1,1} + A_{i,1})x_1 + (cA_{1,2} + A_{i,2})x_2 + \ldots (cA_{1,n-1} + A_{i,n-1})x_{n-1} + 0x_n = cb_1 + b_i$$

The n-th term is therefore eliminated on the i-th equation.

Similarly, the process can be applied to the $(n-1)$-th term. Use the new second equation to eliminate the $n-1$-th term on the third to the last equation. Iteratively applying the elimination $n-1$ times produces an new A' and b' where terms $A'_{i,j}$ with $i > j$ are all zero. In particular, the n-th equation will be a simple $A'_{n,1}x_1 = b'_n$. With x_1 known, the $n-1$-th equation also only has one unknown variable x_2. Iteratively apply the substitution backwards from the n-th equation to the first recovers all of x. The whole process runs with only $O(n^3)$ arithmetic operations.

This method is straight-foward and easy to understand. However, with the usage of division, we can not really solve with method in the world of modulo. Or, do we?

In fact, while division is usually not intuitive in the world of modulo, we can *define* the division in a way. When we think about the definition of a division $\frac{a}{b} = x$, it is actually finding x such that $x \times b = a$. With this in mind, in the world of modulo, we can also try to find x such that $x \times b \equiv a \pmod{m}$. In order to do this, we have to utilize Fermat's Little Theorem.

Fermat's Little Theorem states that for a prime number p and any integer $a, (1 \leq a < p)$, $a^p \equiv a \pmod{p}$. With that in mind, we also know that $a^{p-1} \equiv 1 \pmod{p}$. Now, to find x such that $x \times b \equiv a \pmod{p}$, we can prove that $x = a \times b^{p-2} \pmod{p}$ is a valid (and in fact, the unique) solution.

The final question left to solve this problem is how to calculate $b^{p-2} \pmod{p}$. Doing the calculation naively costs $O(p)$ time, which is too slow (m can be as large as 10^9 in this problem)! This can be done with the technique of exponentiation by squaring.

This technique works with any general exponential calculation with an integer index. In order to calculate a^b, we can calculate $a^0, a^1, a^2, a^4, a^8, \ldots, a^{2^k}$ where $k = \lfloor \log_2 b \rfloor$. Then, since we can decompose b into a sum of powers of 2, we can calculate a^b by multiplying the appropriate powers of a. This can be done in $O(\log b)$ time. Applying modulo to this technique does not change the complexity. Therefore, we can solve the system of equations problem in $O(n^3 \log m)$ time. Now, can you solve it?

Input

The first line of input contains two integers n, m.

The next n lines describe the equations. Each line contains $n+1$ integers: the first n integers are the coefficients $A_{i,1}, A_{i,2}, \ldots, A_{i,n}$ of the i-th equation, and the last integer is b_i. Each line represents an equation $A_{i,1}x_1 + A_{i,2}x_2 + \cdots + A_{i,n}x_n = b_i$.

- $1 \leq n \leq 100$
- $2 \leq m \leq 10^9$, m is a prime number.
- $\forall i, j, 0 \leq A_{i,j} < m$
- $\forall i, 0 \leq b_i < m$

Output

If the system has a unique solution, output n integers on a single line, representing the values of $x_1, x_2, \ldots, x_n$ modulo m.

If the system has no solution, output `NO SOLUTION`.

If the system has more than one solution, output `MORE THAN ONE SOLUTION`.

Sample Input 1

```
3 998244353
1 1 0 2
0 1 0 1
0 1 2 2
```

Sample Output 1

```
1 1 499122177
```

Sample Input 2

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

Sample Output 2

```
6 2
```

Sample Input 3

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

Sample Output 3

```
NO SOLUTION
```

Sample Input 4

```
3 7
1 2 0 3
2 4 0 6
0 0 1 4
```

Sample Output 4

```
MORE THAN ONE SOLUTION
```

Sample Input 5

```
1 7
3 5
```

Sample Output 5

```
4
```

This page is intentionally left blank.

Q. Train Transportation System

Problem ID: train

”Oddsparks: An Automation Adventure” is an open-world crafting survival game. In the game, you can collect various resources, unleash your creativity to design and build automated workshops, and enjoy the fun of automation.

While playing the game, Xiao Yi realized that in order for his workshop to achieve automation, he needed a constant supply of a resource A to appear at location Y. However, resource A is only produced infinitely at a very distant location X. To solve this issue, Xiao Yi decided to build a circular railway, so that a train can transport resource A from location X to location Y, and then return to X to repeat the process.

However, a new problem soon arose. Xiao Yi discovered that, in addition to resource A, he also needed another resource B, which is only produced infinitely at location X', to constantly appear at location Y'. Due to limited materials, Xiao Yi can only afford one train and one looped railway, and the train can only carry one type of resource at a time. Can you help design a circular railway such that both resources can be delivered as required?

Formally, on an $N \times M$ grid, resource A must be transported from (x_{a_1}, y_{a_1}) to (x_{a_2}, y_{a_2}), and resource B must be transported from (x_{b_1}, y_{b_1}) to (x_{b_2}, y_{b_2}). A valid circular railway $R = \{(x_1, y_1), (x_2, y_2), ..., (x_{|R|}, y_{|R|})\}$ must satisfy:

- $1 \le x_i \le N, 1 \le y_i \le M$ for all $1 \le i \le |R|$
- $\forall 1 \le i, j \le |R|, i \ne j, (x_i, y_i) \ne (x_j, y_j)$
- (x_i, y_i) and (x_{i+1}, y_{i+1}) are adjacent, $1 \le i < |R|$

- $(x_{|R|}, y_{|R|})$ and (x_1, y_1) are adjacent
- $\{(x_{a_1}, y_{a_1}), (x_{a_2}, y_{a_2}), (x_{b_1}, y_{b_1}), (x_{b_2}, y_{b_2})\} \subseteq R$

Note that adjacent here refers to four-directional adjacent, meaning (x_i, y_i) and (x_j, y_j) are adjacent if and only if $|x_i - x_j| + |y_i - y_j| = 1$.

Additionally, if the train starts from (x_{a_1}, y_{a_1}) and follows the circular railway, it must visit the positions in the following order: $(x_{a_2}, y_{a_2}), (x_{b_1}, y_{b_1}), (x_{b_2}, y_{b_2})$

It is guaranteed that with the given grid size and positions, there is at least one valid circular railway that satisfies all conditions. If multiple solutions exist, you may output any one of them.

Input

The first line contains two positive integers N and M, representing the size of the grid.

The next four lines each contain two positive integers, representing the positions $(x_{a_1}, y_{a_1}), (x_{a_2}, y_{a_2}), (x_{b_1}, y_{b_1}), (x_{b_2}, y_{b_2})$. And it is guaranteed that the four lines represent distinct coordinates.

- $2 \leq N, M \leq 5$
- $1 \leq x_{a_1}, x_{a_2}, x_{b_1}, x_{b_2} \leq N$
- $1 \leq y_{a_1}, y_{a_2}, y_{b_1}, y_{b_2} \leq M$

Output

The first line should contain a positive integer $|R|$, representing the length of the circular railway R.

The next $|R|$ lines should each contain two positive integers x_i, y_i, representing the positions in the circular railway R.

Sample Input 1

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

Sample Output 1

```
8
1 1
1 2
1 3
2 3
3 3
3 2
3 1
2 1
```

Sample Input 2

```
5 5
2 2
4 4
2 4
4 2
```

Sample Output 2

```
20
2 2
1 2
1 1
2 1
3 1
4 1
5 1
5 2
5 3
5 4
4 4
3 4
2 4
1 4
1 3
2 3
3 3
4 3
4 2
3 2
```

This page is intentionally left blank.

R. Secret Theft

Problem ID: intel

As a member of an elite special forces unit deployed deep in enemy territory, your mission is to infiltrate an enemy area, retrieve critical intel, and safely evacuate:

- Start from your base, marked as `S`.
- Reach the intel point, marked as `T`, to retrieve critical data.
- Proceed to the evacuation point, marked as `E`, for extraction.

The battlefield is laid out as a rectangular maze consisting of walls and open spaces. You may move only one cell at a time in one of the four cardinal directions (up, down, left, right), and only through open spaces. Diagonal movement and passing through walls are not allowed.

Your task is to determine the minimum number of steps required to complete the full mission path from base (`S`), through intel point (`T`), and then to evacuation point (`E`). If it's impossible, report that the mission cannot be completed.

Input

The first line contains two space-separated integers n and m, the number of rows and columns in the maze.

Each of the following n lines contains a string of length m, representing one row of the maze, where:

- `#` represents wall
- `.` represents open space
- `S` represents base/start point
- `T` represents intel point
- `E` represents evacuation/exit point

- $4 \leq n, m \leq 1000$.
- It is guaranteed that there is exactly one base (`S`), one intel point (`T`), and one evacuation point (`E`) in the maze.
- The outermost border of the maze consists entirely of walls (`#`).

Output

Output one integer in a line, the minimum number of steps required to go from base point (S) passing through intel point (T) to evacuation point (E). If it's impossible, print -1.

Sample Input 1

```
5 7
#######
#S...E#
###.#.#
#T....#
#######
```

Sample Output 1

```
12
```

S. Time Split

Problem ID: time-split

Video games are one of the most dominant forms of entertainment in the 21st century and speedrunning is absolutely a popular way of enjoying these digital games. In the domain of speedrunning, the time split tracker is arguably the most indispensable tool for speedrunning. It records the amount of time spent in each segment, which is usually divided according to progressions or milestones in the game, for players to reflect on and improve the runs.

Nathan is an astounding rising star in the speedrunning community. In addition, he owns a time split tracker named Splee. Splee records time in the format of `hh:mm:ss.sss`, where

- `hh` is a two-digit number indicating the hours spent,
- `mm` is a two-digit number indicating the minutes spent,
- `ss.sss` is a number with two digits before the decimal point and three digits after the decimal point, indicating the seconds spent, and
- leading and trailing zeroes are added to ensure each component matches the required digits.
- The hour part should be between 0 and 99, the minute part should be between 0 and 59, and the second part should be between 0 and 59.999.

In addition to that, there are two additional rules for convenience:

- Any prefix containing only zero and colon can be omitted. However, the string must not start with a colon.
- After the decimal point, the trailing zero can be omitted. If there is no number after the decimal place, the decimal point can also be dropped. However, a single decimal point or an empty string is not considered a valid time notation.

For example, 1 hour and 5 seconds can be written as `01:00:05.000`, `1:00:05.000`, or `1:00:05`. One-fifth of a second can be written in the full form, `00:00:00.200`, or in a very simplistic style: `.2`. However, `:00.200` is not valid in this format. To indicate no time at all, either `0`, `.0` or `0.` is allowed, but `.` and an empty string are not valid.

During Nathan's latest world record attempt, one of the segments was lost in the process! Fortunately, he still has the total speedrun time and every other segment recorded in the form that Splee recognizes. Can you implement a program that recovers the lost segment with the remaining information?

Input

The first line of the input contains a number n indicating the total number of the segments. In the following n lines, each line either contains a valid time, representing the time of the segment, or a single question mark character (?) indicating the time of the segment is lost. Finally, there is one more line containing a valid time, which represents the total time of the whole attempt.

- $1 \leq n \leq 100$
- There is exactly one line containing a single question mark character (?).
- The sum of all time of remaining segments will not exceed the total time.

Output

Output a string in a line denoting the lost segment time in the format specified in the statement. If there are multiple valid representations, any of them will be considered correct.

Sample Input 1

```
7
1:39
0:23
1:56
0:02
0:44
0:58
?
6:50
```

Sample Output 1

```
1:08
```

Sample Input 2

```
6
?
10
.100
00.010
0:01:00.
00:00:00.001
1:11.111
```

Sample Output 2

```
00:00:01.000
```

T. Delivery Drone Range Optimization

Problem ID: deliver

You are designing a scheduling system for a company that provides drone delivery services. Each customer has a list of scheduled drone departure times, which indicate when drones can leave to make deliveries.

To coordinate drone operations efficiently, your task is to determine a global time range $[l, r]$ such that each customer has at least one scheduled departure time within this range.

Find the smallest possible range $[l, r]$. If multiple ranges have the same length, choose the one with the smallest l.

Input

The first line contains an integer k, indicating the number of customers. The i-th line of the following k lines contains an integer n_i, the number of scheduled departure times for customer i, followed by n_i distinct integers representing the departure times.

- $1 \leq k \leq 10^5$
- $1 \leq n_i \leq 10^5$
- $\sum_{i=1}^{k} n_i \leq 10^5$
- Each departure time is a positive integer not exceeding 10^9.

Output

Output two integers l and r: the smallest range $[l, r]$ such that each customer has at least one departure time within this range.

Sample Input 1

```
3
5 4 10 24 26 15
4 4 9 12 20
5 5 22 30 18 9
```

Sample Output 1

```
4 5
```

This page is intentionally left blank.

U. Portals

Problem ID: portal

You are trapped in a grid-shaped maze of size $N \times M$.

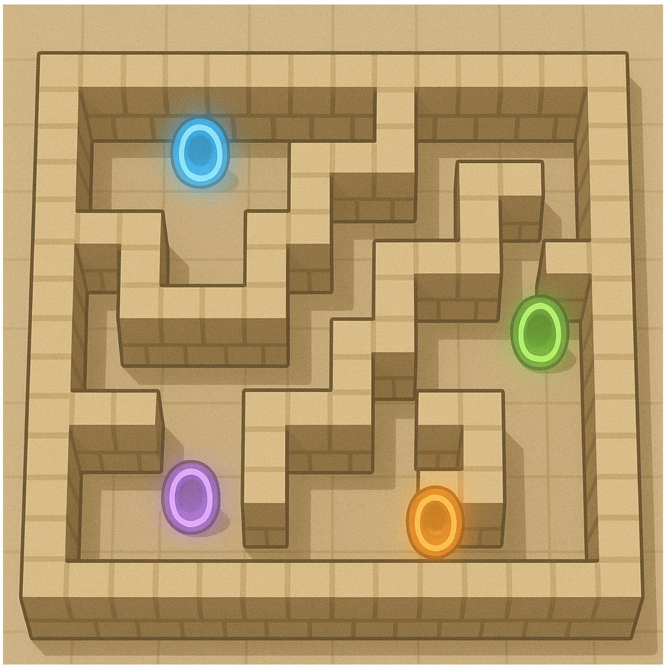


Figure: A maze with some portals.

In this maze, each cell is either empty, a wall, or contains a portal. You begin at the top-left corner (cell $(1, 1)$) and your task is to reach the bottom-right corner (cell (N, M)). You can move to adjacent cells in two directions: right or down, as long as that cell is not a wall. When you enter a cell with a portal, you can choose to enter the portal and get teleported to any cell located further right and down (or on the same row or column) that is not a wall. More specifically, if you are at cell (x, y) with a portal, you can choose to walk normally or teleport to any cell (x', y') such that $x' \geq x$ and $y' \geq y$ and that the cell (x', y') is not a wall.

There are also K treasure boxes in the maze in some non-wall cells. When you are at a cell with a treasure box, you will open the box and receive the value of the box. However, the box might be cursed so you might get a negative value from it. When you enter a cell with a treasure box, you will have to open the box even if you rather not.

Input

The first line of input contains three integers N, M, and K.

The next N lines each contains a string of M characters. The j-th character of the i-th line represents the cell at the (i, j)-th position. The characters may be `.`, `x`, or `p`.

The next K lines each contains three integers x_i, y_i, and v_i. It represents that there is a treasure box at the (x_i, y_i)-th cell with value v_i.

- $1 \leq N, M \leq 1000$
- $0 \leq K \leq N \times M$
- $1 \leq x_i \leq N$
- $1 \leq y_i \leq M$
- $-10^9 \leq v_i \leq 10^9$
- It is guaranteed that all (x_i, y_i) are distinct and not a wall.
- It is guaranteed that $(1, 1)$ and (N, M) are not a wall.
- It is guaranteed that there exists at least one path from $(1, 1)$ to (N, M).

Output

The output contains one integer which is the maximum value you can get.

Sample Input 1

```
3 3 3
...
.x.
...
1 3 2
3 1 -5
3 2 8
```

Sample Output 1

```
3
```

Sample Input 2

```
1 4 2
.pp.
1 3 -1
1 4 2
```

Sample Output 2

```
2
```

This page is intentionally left blank.

V. The God of Mathematics

Problem ID: math

Fysty, in ancient Greek mythology, is the god in charge of prime factorization. He governs the relationships between all numerical ratios in the universe, ensuring that every number remains in its rightful form.

Today, a sudden imbalance has disrupted the harmony among the numbers in the universe, causing chaos. To prevent things from worsening, Fysty must use his divine powers to transform a particular number X into another number Y.

Fysty has two specific ways of using his powers. He can multiply a number x by a prime factor a. If the number x currently has exponent b for that prime a, then this operation costs $b + 1$ units of mathematical power. For example, to multiply $2^6 = 64$ by 2 would cost 7 units of mathematical power. He can also divide a number x by one of its prime factors a. If the number x currently has exponent b for that prime a, then this operation costs b units of mathematical power. For example, to divide $2^7 = 128$ by 2 would cost 7 mathematical power.

Furthermore, since 42 is the ultimate answer to life, the universe, and everything, Fysty cannot transform the number into anything **less than** 42. And to maintain harmony among numbers, at no point during the transformation can the number become larger than the original X.

Can you help Fysty calculate the minimum amount of mathematical power needed to complete this task? Specifically, if the task cannot be completed, output -1.

Input

A single line with two integers X and Y.

- $42 \leq Y \leq X \leq 2 \times 10^5$

Output

Output a single integer representing the minimum total mathematical power Fysty must use to complete this transformation. If the task cannot be completed, output -1.

Sample Input 1

```
864 180
```

Sample Output 1

```
16
```

Sample Input 2

```
1000 500
```

Sample Output 2

```
3
```

W. Fruit Kart

Problem ID: fruit-kart

Fruit Kart is thrilling racing game that supports up to thousands of player competing online simultaneously. The only major flaw of this game is that it only has the free-for-all mode, which is quite disappointing for the players seeking team match possibilities. To solve this, the *Fruit Kart* competitve community have come up with a solution for identifying a team:

- Each member of a team should use the same team prefix in their in-game name.
- The maximum length common prefix of every player in a team is called the *team tag*.
- No two team should have team tags starting with the same character.

Even though this method is intuitive, computing the scores and making the current ranking are not easy tasks. To support other incidents common in the online races, such as disconnections and rule violations, the community have therefore decided the following format and rules for recording and making the ranking of the match.

- Each entry of the match is an in-game name followed by a score record. Each player in the match has exactly one entry associated with the player.
- Each record is an expression, with numbers, plus sign and negative sign. There will not be consecutive plus or negative signs.
  - When a number is preceded by a plus sign or no sign at all, it contributes to the raw score of the player. The raw score of the player is the sum of all such numbers.
  - When a number is preceded by a minus sign, it contributes to the penalty score of the player. The penalty score of the player is the sum of all such negative numbers.
- The total score of a player is the raw scores of the player plus the penalty scores of the player.
- The total score of a team is the sum of total scores of players.
- Both teams and players are ranked according to the total score. The higher scoring one receives a higher rank. In case of a tie, all tied players or teams share the same rank and subsequent ranks are skipped accordingly.

Even with this format and rule set, sometimes it is still unclear that who is the winner from the record. This is where the table maker comes into play. The table maker computes the scores and team tags, and print them into 7 columns.

- In the table, each row is a player, the rows are first sorted according to the team ranks and then the individual ranks. If there is a tie, the team or player listed earlier in the record will be listed earlier in the table.
- Each column is of width the longest cell in the column.
- Each cell is left-aligned and padded with space to the width of the column, except that no trailing space will be added in the last column.
- In addition, one space is added between every pair of adjacent columns.
- The first, second, and third column contains the rank, the team tag, and the total score of the team, which only appears in the first row of each team.
- The fourth, fifth, sixth, seventh column contains the in-game name, raw score, penalty score, and the the individual rank of the player.
- In first and the last column, the ranks is represented by #r where r is the rank.
- In the sixth column, the cell is empty if the penalty score is zero, or a negative integer surrounded by brackets if there is.

With this table, all statistics of the match become apparent. Can you help the community to make a robust table maker?

Input

The first line of the input contains a number n indicating the total number of the entries. In the following n lines, each line contains an entry in the format described in the statments.

- $1 \leq n \leq 2000$
- There are at least two teams.
- The in-game name of each player contains only printable ASCII with code between 33 and 126 (printable characters excluding the space).
- The length of the in-game name of each player is between 1 and 2000.
- The absolute value of the raw and penalty score of a player does not exceed 10^9.
- The length of each score record is between 1 and 2000.
- There will not be leading zero in the score records. Please note that single zero can still occur.

Output

Output the table in the format described in the statments.

Sample Input 1

```
12
RRLight 74
SS-Markus 97
#always 56
SS-Charlie 106
figure 68
#never 109
==GOAT== 72
RRDark 94
==FOUL== 26+20+16-10
discuss 84
figlover 89
disadvantage 83
```

Sample Output 1

```
#1 SS- 203 SS-Charlie   106       #2
           SS-Markus    97        #3
#2 RR  168 RRDark       94        #4
           RRLight      74        #8
#3 dis 167 discuss      84        #6
           disadvantage 83        #7
#4 #   165 #never       109       #1
           #always      56        #11
#5 fig 157 figlover     89        #5
           figure       68        #10
#6 ==  124 ==GOAT==     72        #9
           ==FOUL==     62  (-10) #12
```

Sample Input 2

```
12
(:P) 0+0+0-0-0-0
(OAO) 1+0+0-0+1
average 27
Annoying 33
Astute 33
SoloShow 10+10+12+100
aversion 12
(((.o.))) 15+15+16
Airplane 33
Awful 33
(._.)'' 12+14
averylongname 26+20
```

Sample Output 2

```
#1 A        132 Annoying      33   #4
                Astute        33   #4
                Airplane      33   #4
                Awful         33   #4
#1 SoloShow 132 SoloShow      132  #1
#3 aver     85  averylongname 46   #2
                average       27   #8
                aversion      12   #10
#4 (        74  (((.o.)))     46   #2
                (._.)''       26   #9
                (OAO)         2    #11
                (:P)          0    #12
```

Sample Input 3

```
3
A -1
A -1
B -2
```

Sample Output 3

```
#1 A -2 A 0 (-1) #1
        A 0 (-1) #1
#1 B -2 B 0 (-2) #3
```

X. Forest of Equilibrium

Problem ID: trees

In the enchanted land of Numaria, the ancient Elven Forest Council seeks to restore balance to the magical forest. Along a straight forest path (a number line), there are already n ancient trees, each rooted at a unique integer position.

However, the elves have noticed that some gaps between trees are too wide, disturbing the natural flow of magical energy. To restore harmony, they plan to plant up to k new trees at integer points strictly between existing ones (not before the first or after the last tree) to improve the uniformity of spacing. In other words, the goal is to minimize the maximum distance between any two adjacent trees by optimally placing the new trees.

Your task is to determine the minimum possible value of the maximum distance between two adjacent trees, after optimally planting up to k new trees.

Input

The first line contains two integers n and k. The second line contains n space-separated integers $p_1, p_2, \ldots, p_n$, representing the positions of the existing trees on the number line.

- $2 \le n \le 10^5$
- $0 \le k \le 10^9$
- $-10^9 \le p_1 < p_2 < \ldots < p_n \le 10^9$

Output

Output a single line containing one integer representing the minimum possible value of the maximum distance between any two adjacent trees after planting up to k new trees.

Sample Input 1

```
2 1
-8 12
```

Sample Output 1

```
10
```

Sample Input 2

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

Sample Output 2

```
1
```

Sample Input 3

```
5 4
-20 -18 -4 6 9
```

Sample Output 3

```
5
```

Y. Painter

Problem ID: painter

LittleCube has an empty rectangular grid. The grid contains $w \times h$ cells and each of the cells is an unit length square. If we take the lower left corner as the origin and make a 2D coordinate over the grid, the grid occupies the area where $0 \leq x \leq w$ and $0 \leq y \leq h$. We denote the region of $u \leq x \leq u + 1$ and $v \leq y \leq v + 1$ as the cell (u, v).

LittleCube created a masterpiece by simply drawing a lot of lines on top. In each step, he picked two cells (x_1, y_1) and (x_2, y_2) and drew a line between the center of the two cells. Every cell intersected with the line was filled with the color of the line, overwriting the previous color if there was. In case of $(x_1, y_1) = (x_2, y_2)$, the line degenerates into a point and he would only paint the cell (x_1, y_1).

However, LittleCube has lost the drawing. Fortunately, he still holds the original record of all instructions. Can you recreate the masterpiece for him?

Input

The first line of the input contains three integers n, w, h, indicating the total number of the instructions, the width of the grid, and the height of the grid. In the following n lines, each line contains four integers and an uppercase Latin character, x_1, y_1, x_2, y_2, c, indicating LittleCube drew a line between the center of cells (x_1, y_1) and (x_2, y_2) with color c. The instructions are given in order.

- $1 \leq n, w, h \leq 5000$
- For all instructions, $0 \leq x_1, x_2 < h$ and $0 \leq y_1, y_2 < w$.

Output

You should output h lines, each line with w characters (excluding the new line character). In the i-th line, the j-th charater (both counting from 1) should be the color of the cell $(h - i, j - 1)$. If the cell is never colored, output a single dot (`.`) instead for that cell.

Sample Input 1

```
5 10 10
0 0 9 9 A
1 2 3 4 B
1 8 7 1 C
2 8 9 8 D
8 1 8 1 E
```

Sample Output 1

```
........AA
.CDDDDDDDD
.CC...AAA.
..CC.AAA..
...CCAA...
..BBCC....
.BBBACC...
.BBA..CC..
AAA....CE.
AA........
```

Z. Cutscenes

Problem ID: cutscenes

Do you remember the *Fruit Kart* racing game from some previous problem? Now, LittleCube is trying yet another way to enjoy it: *all-tracks speedrun*!

In this game, tracks are laid on an $n \times m$ grid where each cell is a track. That is to say, there is a track (i, j) for every $1 \leq i \leq n$ and $1 \leq j \leq m$. *All-tracks speedrun* is as simple as it is: play every track at least once with the shortest time. However, it is not just racing that takes time. LittleCube found that sometimes when you play on two nearby tracks, there will be a transition cutscene showing the player driving from one track to another, which is not only boring for speedrunners, but also wasting a great amount of time! Therefore, the order of playing these tracks actually matters substaintially in the results!

Besides practicing the driving skill, LittleCube also tried to the game's mechanics. After some experiments, he figured out that the game has exactly k templates of cutscenes. Each template can be described with three parameters: $\Delta x_i, \Delta y_i$, and t_i. A cutscene template will be played during the transition from any track (x, y) to $(x + \Delta x_i, y + \Delta y_i)$, taking t_i unit of time regardless of the starting or ending track.

LittleCube wants to minimize the total unit of time spent watching cutscenes while each track is played *exactly once* since driving on tracks is nevertheless the most time-consuming part. Can you help him find one such route?

Input

The first line of the input contains two integers n, m, indicating the dimension of the grid. The second line of the input contains an integer k, indicating the number of cutscene templates. Next, k lines follow, where the i-th of the k lines contains three integers $\Delta x_i, \Delta y_i, t_i$.

- $1 \leq n, m \leq 1000$
- $0 \leq k \leq 24$
- $0 \leq |\Delta x_i|, |\Delta y_i| \leq 2$
- $1 \leq t_i \leq 10^8$
- All $(\Delta x_i, \Delta y_i)$ are pairwise distinct and not identically $(0, 0)$.

Output

In the first line, output an integer indicating the minimum units of time spent watching cutscenes over all routes. In the second line, output $2 \times n \times m$ integers $x_1, y_1, x_2, y_2 \ldots, x_{nm}, y_{nm}$ indicating that one of such routes is playing the tracks in the order of $(x_1, y_1), (x_2, y_2), \ldots (x_{nm}, y_{nm})$.

Sample Input 1

```
5 2
1
1 0 10
```

Sample Output 1

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

Sample Input 2

```
2 2
4
0 1 10
1 0 20
-1 0 30
0 -1 40
```

Sample Output 2

```
10
1 2 2 1 2 2 1 1
```

Sample Input 3

```
4 1
4
-2 0 30
-1 0 50
1 0 70
2 0 40
```

Sample Output 3

```
60
3 1 1 1 4 1 2 1
```

Sample Input 4

```
2 3
7
1 0 100
-1 -1 10
0 -1 100
1 -1 100
-1 1 100
0 1 100
1 1 100
```

Sample Output 4

```
20
2 2 1 1 1 3 2 1 2 3 1 2
```

This page is intentionally left blank.