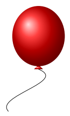
Problem A. Alpha Country

Ballon:	
Time limit:	1 seconds
Memory limit:	512 megabytes

In Alpha country there are n islands numbered from 1 to n. The islands are connected by a one-way bridge (island i can only go to island $i + 1$). On each island you arrive, you can receive a bonus or must pay a fine m_i dollars (m_i is a positive number represents the amount of money you will receive and m_i is a negative number represents the amount of money you will must pay a fine) (The current your amount of money can be negative).

Tuan is given two times to use magic by a magician, one time can teleport to any island (use the first time go to Alpha country), go to the next island sequentially and one time can return to his home anytime. However, He will have to send back to the magician an amount equal to the largest amount was collected on an island that he arrived. **Tuan** will use magic optimally to earn the maximum amount of money.

Print the maximum possible amount of money **Tuan** can earn.

Input

The first line contains a single integer n safety $1 \leq n \leq 10^5$ - the number of islands in Alpha country.

The second line contains n integers $m_1, m_2, \ldots, m_n$ safety $-500 \leq m_i \leq 500$ - the amount of money you can receive a bonus or pay a fine in island i.

Output

Print a single integer – the maximum possible amount of money **Tuan** can earn.

Examples

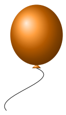
standard input	standard output
5 6 -5 7 3 -2	4
8 -1 5 -4 3 4 6 -10 4	8

Note

In the first test case, **Tuan** teleports to the $1st$ island, go to $2nd, 3rd, 4th$ island and go to home in $4th$ island. He earns $(6 - 5 + 7 + 3) = 11$ dollars and give magician 7 dollars, so he can earn 4 dollars.

In the second test case, **Tuan** teleports to $2nd$ island and exit in $6th$ island and he earn 8 dollars

Problem B. Ocean Club

Ballon:	
Time limit:	1 seconds
Memory limit:	512 megabytes

Bi is very fond of the Ocean series on HBO. Because she liked it so much, when she was in her class, she created her own Ocean club. To practice skills like the characters in the movie, **Bi** gave a math problem to its members practice as follows: Bi's Ocean club has n members with a list of home address numbers $A = \{a_1, a_2, \ldots, a_n\}$. The association's communication rules are from small house address numbers to larger house address numbers and co-primes. **Bi** requires members to calculate quickly to transmit information from a house has an address a_i to a_j number another house address and must pass through $k + 1$ people, how many ways are there?

The members of Bi's Ocean Club are not good at it yet, so **Bi** ask you to help calculate.

Input

The first line contains n is the number of members safety $1 \leq n \leq 100$.

The second line contains n integers is distinct $a_1, a_2, \ldots, a_n$ safety $1 \leq a_i \leq 10^5$ - the list address.

The third line contains one integer Q safety $1 \leq Q \leq 10^5$ — the number of query Bi will do.

The next Q lines contains three integers a_i, a_j, k safety $a_i \leq a_j; a_i, a_j \in A; 1 \leq k \leq n$.

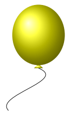
Output

Output Q lines, the $i-th$ line should contain the $i-th$ query result. Since the number of result is large, print it modulo 2023.

Examples

standard input	standard output
7 2 3 4 5 6 7 8 1 2 8 3	3

Problem C. Greatest common divisor



Ballon:	
Time limit:	1 seconds
Memory limit:	512 megabytes

Given a sequence of numbers A consisting of n positive integers $1 \le n \le 100$) the value of ith number is a_i, $(1 \le a_i \le 70)$.

Considering all non-empty subsets of the sequence A, we calculate the greatest common divisor of the numbers. What is the sum of all common divisors of all subsets? Knowing the result is very large, we only need to output the result when taking the remainder for $10^9 + 7$.

Input

The first line contains a positive integer T, $(1 \le T \le 40)$ which is the number of data sets in the problem.

The next T datasets are each organized as follows:

- The first line contains a positive integer n.

- The second line contains n positive integers, the ith value is a_i

Output

The result of each test set includes a positive integer representing the result of the problem.

Examples

standard input	standard output
2 5 1 1 1 1 1 3 1 2 3	31 10

Note

Testcase 1: Because the numbers are the same, every set has a greatest common divisor of 1. There are $25 - 1 = 31$ non-empty sets, so the sum of the common divisor is 31.

Testcase 2: The non-empty sets are $\{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}$ with corresponding greatest common divisors are $1, 2, 3, 1, 1, 1, 1$ and the sum is $1 + 2 + 3 + 1 + 1 + 1 + 1 = 10$.

Problem D. Boom

Ballon:	
Time limit:	1 seconds
Memory limit:	512 megabytes

Alice is playing another game called BOOM. The game has a rectangular map of size $N \times M$. Each 1×1 cell is a wall, an enemy, or an empty cell. In this game, a bomb has a horizontal explosion range with radius V. In other words, if the bomb is placed at cell (i, j), its explosion range is from the cell $(i, j - V)$ to the cell $(i, j + V)$. It is not allowed to place bombs at the wall cells, it is allowed to place bombs in the remaining cells. The bomb explodes at its placed cell and spreads to both horizontal sides, until it reaches the end of its explosion range or encounters a wall. Note that the wall will not be destroyed by bombs.

In this game, **Alice** will carry exactly K bombs. Therefore, before entering the map, **Alice** must buy K bombs with an explosion range of V. To save in-game money, **Alice** will buy exactly K bombs with the minimum V but can still destroy all enemies in the map.

Please help **Alice** to calculate the minimum V.

Input

The first line contains three positive integers N, M, K safety $N \times M, K \leq 10^5$.

In the next N lines, each line contains M characters of the following characters: '#' is a wall, '.' is an empty cell, 'x' is an enemy.

Output

Output the minimum V so that **Alice** can place K bombs to destroy all enemies. If there is no answer, output -1.

Examples

standard input	standard output
4 6 5 #x..#. #.x.x# ..#.xx x....x	1
2 3 10 xxx xxx	0
2 2 1 x. x.	-1

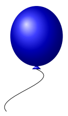
Note

Test case 1: With $V = 1$, she can place 5 bombs in cells: $(1, 2), (2, 4), (3, 5), (4, 1), (4, 6)$. This is the minimum V.

Test case 2: She can place bombs in all cells in the map.

Test case 3: She cannot destroy all enemies with 1 bomb.

Problem E. Boxes and Flags

Ballon: 
Time limit: 1 seconds
Memory limit: 512 megabytes

This is an interactive problem.
An interactive problem require communication between submission and the interactor via their stdin/stdout. Submission's stdout is interactor stdin and vice versa.

Alice and **Bob** are partaking in a puzzle game with a prize. In this game, there will be 2 players **Alice** and **Bob** and there will be a Gamemaster. There are two types of room: the Waiting Room and the Playing Room. Initially, they are both in the Waiting Room, and the Gamemaster is in the Playing Room.

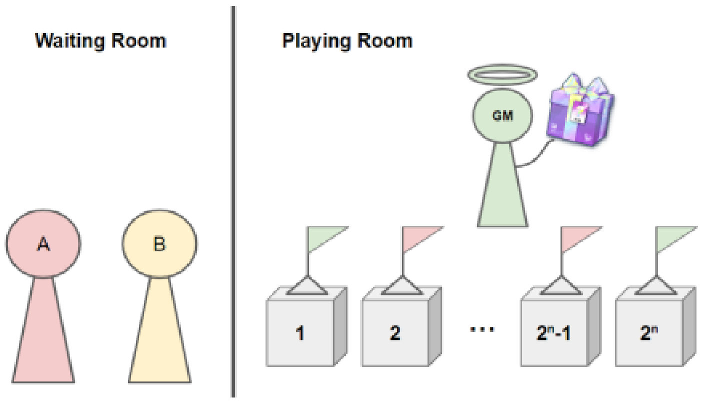


Figure 1: Initial setting of the game

The game goes as follow:

1. In the Playing Room, there are 2^n identical empty boxes, numbered from $1, 2, \ldots$ and so forth. On top of each box there is a flag with only two colors (of Green and Red). Initially, all flags have their color randomed.

2. Firstly, **Alice** will come in, see the state of all flags, see the Gamemaster put the reward in a random box. After that, **Alice** must pick one flag and change its color (from Green to Red, or Red to Green).

3. Then, **Alice** will leave the room without seeing **Bob**. And **Bob** will enter the Playing Room. He must somehow with only 1 guess, correctly guess the box which contains the prize, just by the colors of the flag.

After the Gamemaster finishes explaining the rules to **Alice** and **Bob**, they all think this game is a scam. But the Gamemaster keeps persisting that it is possible, you just need a good strategy beforehand. Would you help them discuss the strategy as well?

Also, the Game Master really does not want you to win by luck, so instead of 1 Playing Room, there will be T consecutive Playing Rooms. **Alice** after finish with Playing Room i will move out of that room, and wait for **Bob** to finish his turn at Room i. After that, if **Bob** guesses correctly, they will repeat the game in the next room, the only difference that could occur are the flags and the position of the prize.

How To Interact:

Please check the Example section for better understanding. From the problem statement, for better clarity, the interaction is divided into 4 phases:

1. **Phase 1**: The interactor will prepare 2^n boxes and flags, and output n for the participant to read. $(1 \leq n \leq 6)$.

2. **Phase 2**: Participant then prints 2^n lines. $i - th$ line contains a mathematical expression (details are below this section [1]). The $i - th$ expression describes to **Bob** that IF evaluation of that expression is true, the prize should be in $i - th$ box. (aka. This is the phase where **Alice** gives **Bob** the strategy before heading into the Playing Rooms. The strategy is only given once, and is used by **Bob** throughout T playing rooms).

3. **Phase 3**: The interactor output T, announce the amount of Playing Rooms there are. $(1 \leq T \leq 100)$

4. **Phase 4**: For each room i:

a. Firstly, the interactor output on the same line 2^n values of 0 and 1, $j - th$ value describe the colors of the $j - th$ flag. They are separated by a space.

b. Next, the interactor will output another line containing one integer, which is the position that the Gamemaster hid the prize. (aka. **Alice** enters Playing Room $i - th$ and observes where the gamemaster put the prize). Now, the participant must output an integer value *cpos*, represents the position of the flag that **Alice** must change its color. $(1 \leq cpos \leq 2^n)$

Finally, the interactor will check if the color change will help locate the prize according to the strategies given at the start of the interaction. If it's correct, the interactor will output the word **OK** on a single line. Otherwise, it will output **BAD** on a single line, at this point you should terminate your program to avoid undefined behaviors.

[1] For constraints of the mathematical expression:

- Length of expression should not exceed 2506 characters.

- For operands, you can use integers. At every point of calculation, absolute value must not exceed 10^18 to avoid undefined behaviors.

- Other than integer operands, you can also use $[x]$ with x is the index of the box $(1 \leq x \leq 2^n)$ to obtain the value of the flag on top of $x - th$ box. If the color is Green, $[x]$ is 0. If the color is Red $[x]$ is 1.

- For operators, you can use addition $+$, subtraction $-$, multiplication $*$, integer division $/$, and modulo %. For the division behavior or precedence order, it is the same as defined in language C++.

- You can use parenthesis ( ).

- You can use comparisons such as $=$ (equality), $>$ (greater), $<$ (lesser). Which will output result 1 if the comparison was truthful, otherwise 0.

- 0 is considered a boolean value of false. Other integers are true. For every case, there should exist only 1 true and 2^{n-1} false from given expressions. Failure to evaluate expression (such as divided by zero) also leads to a WA.

- It can be proven that a solution exists within the given constraints. Attempts to break constraints will lead to undefined behaviors, WA is high probability.

For example, in your strategies your $25th$ line looks like this:

```
...
([1] + [2] + [3]) * 2 = 6
...
```

Then **Bob** will check the $25 - th$ box if the first, second, and the third flag are all Red.

Examples

Example 1

Interaction (Left is Participant, Right is Interactor)	**Notes**
1	Interactor output n
[1] - [2] $>$ 0 [2] $>$ [1]	Participant prints 2^n strategies
2	Interactor announce T
	Room#1
0 0 1	Flags state and prize position
1	**Alice** flips 1-st position. State 00 is now 10.
OK	Strategies evaluated as: - $1st$ expression: [1]>[2] is true - $2nd$ expression: [2]>[1] is false Then **Bob** should check the 1st box since the 1st expression is the only truthy one, and it's the correct answer.
	Room #2
1 1 2	State and new hidden position for prize for room 2.
1	State 11 is now 01
OK	Strategies evaluated as: - $1st$ expression: [1]>[2] is false - $2nd$ expression: [2]>[1] is true Then **Bob** should check the 2nd box. It is correct.

Example 2

Interaction (Left is Participant, Right is Interactor)	**Notes**
1	Phase 1, the interactor prepares 2^n boxes and outputs n for participants.
([2] = 0) ([2] = 1)	Participant prints 2^n strategies
2	Interactor announce T
	Room #1
0 0 1	- Interactor first print state of all flags. In this case, all have Green colors. - On $2nd$ line, it outputs the position of the prize, which is in the $1st$ box
1	- **Alice** flips colors of the $1st$ flag. State 00 is now 10
OK	- According to the strategies, 10 has $2nd$ flag is a 0, so the prize should be in the $1st$ box - Correct. Interactor outputs OK
	Room #2
0 1 2	State and new hidden position for prize for room 2.
1	State 01 is now 11
OK	$2nd$ cell has value 1, it matches the $2nd$ expression.

Example 3

Interaction (Left is Participant, Right is Interactor)	**Notes**
2	Interactor output n
(([1]+[2]) = 1) (([1]+[3]) = 1) (([1]+[4]) = 1) (([2]+[4]) = 1)	Participant prints 2^n strategies
3	Interactor announce T
Room #1	
1 1 1 1 2	
3	**Alice** flips the color of the $3rd$ flag. State 1111 is now 1101.
OK	According to strategies evaluation the $2nd$ strategy is truthy, it claims that the prize is in $2nd$ box and it was correct. Interactor outputs OK
Room #2	
0 0 0 0 1	State and new hidden position for prize for room 2.
1	State 0000 is now 1000
BAD	This state has more than 1 truthful expression, so **Bob** does not know which to open.
Terminal	

Problem F. Milk Tea

Ballon: 
Time limit: 1 seconds
Memory limit: 512 megabytes

Suika is a big fan of Taiwanese milk tea, she can even drink a big glass of this beverage every single day, sometimes two. After the last academic year, she is awarded as the highest-performing student of the whole school. To celebrate this achievement, her parrents decide to give her a trip to Taiwan so that she can joyfully relax and prepare for the next year.

Suika of course plans to drink as much milk tea as she can while she is in Taiwan. However, if just trying some milk tea is not enough for her. After a long time thinking, she tells her parrents that she will try milk tea in every town she travel. She also has a very clear plan for the towns she will visit. But she is only a little girl and this is the first time she travels to a foreign country without her parrents, so **Suika** just keeps a shortest list of roads to travel between those towns. Suppose there are N cities she wishes to visit, she only know $N-1$ bidirectional roads for simplicity (poor little girl). She numbers those towns from 1 to N and decide that she will visit those in order (from 1 to N). It means, she will visit town 1 and drink a glass of milk tea, go for sightseeing, then visit town 2, drink milk tea again.

To traverse a specific road, she will need to have a valid train ticket. There are 2 types of ticket for travelling, in the $i-th$ train route, single-pass ticket which costs Ci_1 Taiwan dollar, and multi-pass ticket which costs Ci_2 dollar. For each train route, she can decide to buy a single-pass ticket each time she needs to get in, or she might buy a multi-pass ticket once.

Let's help **Suika** to find the smallest amount for Taiwan dollars she will have to spend so that she can fully enjoy her trip.

Input

The first line contains and integer N $(2 \le N \le 10^5)$ for task description.

In the next $N-1$ lines: there are 4 integer number A_i, B_i, Ci_1, Ci_2 which represents that towns A_i and B_i are connected with a train route, with ticket prices Ci_1 and Ci_2. Data safety $(1 \le A_i, B_i \le N, 1 \le Ci_1, Ci_2 \le 100000)$.

Output

A single line output the smallest cost of her travel.

Examples

standard input	standard output
5 1 2 1 3 1 3 1 4 2 4 2 3 3 5 2 3	11
5 1 2 3 5 1 3 2 3 1 4 1 3 1 5 2 3	12

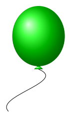
Note

in Testcase1, this sample is straightforward, **Suika** needs to travel 4 routes with $4, 3, 2, 1$ times in order.

Thus, she will buy multi-pass for route 1 and 3, single-pass for route 2 and 4.

Cost: $3 + 3 \times 1 + 3 + 1 \times 2 = 11$

Problem G. Remove or Split

Ballon:	
Time limit:	1 seconds
Memory limit:	512 megabytes

Alice and **Bob** are playing a game with n piles of stones. The $i-th$ pile has a_i stones. **Alice** and **Bob** will play a game alternating turns with **Alice** going first. On a player's turn, they must perform one of two following operations:

1. Choose one pile and remove a positive number of stones.
2. Choose one pile and split it into two non-empty piles.

The first player unable to make a move loses.

Given the starting configuration, determine who will win the game if both players play optimally.

Input

The first line contains one integer n safety $(1 \leq n \leq 10^5)$ - the number of piles.

The second line contains n integers $a_1, a_2, \ldots, a_n$ safety $(1 \leq a_i \leq 10^4)$ - the number of stones in the piles.

Output

Print a single string "Alice" if Alice wins; otherwise, print "Bob" (without double quotes).

Examples

standard input	standard output
1 4	Alice
3 1 2 4	Bob

Note

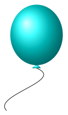
In the first example:

First turn, **Alice** removes 4 stones from the pile, thus there are no stones left, and **Alice** wins.

In the second example:

First turn, **Alice** removes 1 stone from the $3rd$ pile. The game state is $[1, 2, 3]$. Second turn, **Bob** split $3rd$ pile into two piles $[1, 2]$. The game state is $[1, 2, 1, 2]$. Remaining turn, no matter what move **Alice** chooses, **Bob** will repeat exactly, and **Bob** wins.

Problem H. Health check

Ballon: 
Time limit: 1 seconds
Memory limit: 512 megabytes

Chi is a busy person. But because of working too much, he is suffered from some diseases that needed to be examined and the doctor required many tests to be able to judge **Chi**'s health.

After receiving the request, **Chi** looked at the calendar and saw that he had n, $(1 < n < 10^5)$ upcoming free days and **Chi** listed them on paper $a_1, a_2, \ldots, a_n$, $(1 < a_i < 10^9)$ is the distance from today to the next free day. For example, tomorrow is free, then a_1 is 1.

The doctor asked **Chi** to go to the hospital and do m, $(m \leq 10^3)$ tests. He is can only go to the hospital on free days, but can do many different tests on the same day. For each test, **Chi** will need z, $(z < 10^9)$ sets of documents to be able to conclude the disease.

A document is a chart that compares body indicators between any two days to monitor health conditions. A valid document requires that the gap between two test dates must be between x and y days $(1 < x \leq y < 10^9)$ and the hospital only needs enough documents before concluding the disease. without prohibiting the time of any two sets of documents to overlap for a period of time. That means you can use data from a past medical examination day to create multiple sets of documents for the same test.

Input

The first line contains one integer n, m.

The second line contains n increasing non-negative natural numbers represent the upcoming free days of **Chi**.

The next m lines, each line includes 3 numbers: x, y, z. With x, y, z is the requirement that **Chi** must complete z sets of documents, with the condition of each set being the gap between 2 days in the range of $x - y$ days.

Output

Print the earliest date that **Chi** will complete all tests. If that date is not available, return -1.

Examples

standard input	standard output
5 1 1 2 4 5 7 2 3 2	4
8 2 1 4 6 8 12 16 18 20 2 3 4 4 5 4	18

Note

In the first example: **Chi** will make 2 sets of documents. 1 set is to compare day 4 compared to day 1. 1 set is to compare day 4 compared to day 2. Complete at the earliest on day 4.

In the second example: Test 1 will be completed on day 18 at the earliest. Test 2 will be completed on the day 16 at the earliest. Print out 18.

Problem I. Dwayne and Megan

Ballon: 
Time limit: 1 seconds
Memory limit: 512 megabytes

Within the premises of a castle represented as an $N \times M$ grid, two individuals, **D**wayne and **M**egan, reside. The castle has doors marked with '#' symbols and empty spaces marked with '.' symbols. **D**wayne and **M**egan can only move in four directions: up, down, left, and right. Another person, referred to as **X**men, wants to separate **D**wayne and **M**egan by locking all the doors and constructing at most one new door at any empty location within the castle, except for the current positions of **D**wayne or **M**egan. The condition for **X**men to succeed is that there should be no path between **D**wayne and **M**egan that allows them to reach each other.

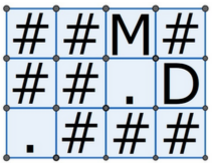


Figure 2: In the given diagram, **X**men can separate **D**wayne and **M**egan by placing a single wall between them.

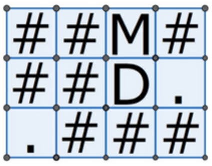


Figure 3: In the given diagram, **X**men cannot separate **D**wayne and **M**egan.

Find the number of pairs of positions for **D**wayne and **M**egan that allow **X**men to successfully divide them. Each square can only contain a single person, so **D**wayne and **M**egan must be in different squares. This means that there will be at least two squares containing '.' for them to occupy.

Input

The first line contains two integers N and M separated by a space, where $1 \leq N, M \leq 500$.

The next N lines each contain a string of length M consisting of '#' and '.', where '#' represents a door and '.' represents an empty space in the castle. It is guaranteed that there will be at least two squares containing '.'

Output

Print an integer representing the number of unique pairs of positions for **D**wayne and **M**egan that allow **X**men to divide them with at most one door.

Examples

standard input	standard output
3 4 ##.# ##.. .###	8
2 3 #.. #.#	2
2 3	0
4 5 #...# #.##. ..##. ..##.	86

Note

In example 1, there are 8 valid pairs of position **D**wayne and **M**egan.

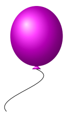
1: $\begin{bmatrix} \# & \# & M & \# \\ \# & \# & . & D \\ . & \# & \# & \# \end{bmatrix}$ 2: $\begin{bmatrix} \# & \# & M & \# \\ \# & \# & . & . \\ D & \# & \# & \# \end{bmatrix}$ 3: $\begin{bmatrix} \# & \# & . & \# \\ \# & \# & M & . \\ D & \# & \# & \# \end{bmatrix}$ 4: $\begin{bmatrix} \# & \# & . & \# \\ \# & \# & . & M \\ D & \# & \# & \# \end{bmatrix}$

5: $\begin{bmatrix} \# & \# & D & \# \\ \# & \# & . & M \\ . & \# & \# & \# \end{bmatrix}$ 6: $\begin{bmatrix} \# & \# & D & \# \\ \# & \# & . & . \\ M & \# & \# & \# \end{bmatrix}$ 7: $\begin{bmatrix} \# & \# & . & \# \\ \# & \# & D & . \\ M & \# & \# & \# \end{bmatrix}$ 8: $\begin{bmatrix} \# & \# & . & \# \\ \# & \# & . & D \\ M & \# & \# & \# \end{bmatrix}$

In example 2, there are 2 valid pairs of positions:

1: $\begin{bmatrix} \# & . & D \\ \# & M & \# \end{bmatrix}$ 2: $\begin{bmatrix} \# & . & M \\ \# & D & \# \end{bmatrix}$

Problem J. First strip

Ballon: 
Time limit: 1 seconds
Memory limit: 512 megabytes

In order to win **Yen Nhi's** heart, **Quang Tri** accepts a brain-tricky challenge from her. **Yen Nhi** draws on the floor a strip of n consecutive squares (called the first strip). The squares are numbered 1 to n from left to right. For each square, **Quang Tri** has to choose either the color red or blue to paint it. Note that all squares must be colored in a certain color, i.e, **Quang Tri** cannot leave any square blank.

Yen Nhi has n pens with red ink and n other pens with blue ink to support **Quang Tri**'s coloring process. Each pen is filled with enough ink to fill exactly a single square with the corresponding color. **Yen Nhi** stipulates that **Quang Tri** initially has 0 points, and if he wants to get i **red-ink** pens, a_i points will be deducted from his score. Similarly, **Quang Tri**'s score will have b_j points deducted if he wants to take j **blue-ink** pens from **Yen Nhi**. Note that $a_0 = b_0 = 0$.

Also, **Yen Nhi** gives **Quang Tri** a special rule to get more points. She has another strip of squares of length m (called the second strip) and all the squares are already colored either red or blue. For each contiguous segment of length m in the first strip whose coloring coincides with the second strip, **Quang Tri** will receive D points. Tri wants to find a way to color the first strip to maximize his final score (so he can make a good impression on **Yen Nhi**). Let's help **Quang Tri** compute the optimal solution!

Note that a segment's coloring is considered "coincided" with another segment's coloring if there does not exist a pair of squares with the same index (from left to right) in both segments but have different color.

Input

The first line contains three integers n, m, and D, $(1 \leq n \leq 2000, 1 \leq m \leq min(n, 10), 1 \leq D \leq 10^6)$. n and m is the length of the first and second strip, respectively. D is the bonus amount **Quang Tri** can obtain for each contiguous segment of the first strip whose coloring coincides with the second strip.

The second line contains n integers $a_1, a_2, \ldots, a_n$, $(1 \leq a_i \leq 10^9)$.

The third line contains n integers $b_1, b_2, \ldots, b_n$, $(1 \leq b_i \leq 10^9)$.

The fourth line contains a string of length m whose each character is either 'R' or 'B' representing that the corresponding square of the second strip is colored with the color red or blue.

Output

Print two lines, the first line contains the maximum final score that **Quang Tri** can obtain, and the second line contains a string of length n representing his coloring solution: $i-th$ character is 'R' if the $i-th$ square in the first strip is colored with red, and is 'B' otherwise. If there are multiple valid answers, you can output any of them.

Examples

standard input	standard output
6 2 1 1 2 3 4 5 6 6 5 4 3 2 1 RR	-1 RRRRRR
6 2 4 1 2 3 4 5 6 6 5 5 3 2 1 RB	4 RBRBRB
6 2 3 1 2 6 4 5 6 6 5 4 3 2 1 RB	1 RBRBBB

Note

In example 1, two possible ways that **Quang Tri** can color the first strip to achieve -1 points:

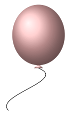
Quang Tri uses 6 red-ink pens. There are 5 contiguous segments whose coloring coincides with the second strip.

Final score $= -a_6 - 0 + 5 * D = -6 + 5 = -1$.

Quang Tri uses 6 blue-ink pens. There are 0 contiguous segments whose coloring coincides with the second strip.

Final score $= -0 - b(6) + 0 * D = -1$.

Problem K. Handmade Billiards Table



Ballon:
Time limit: 1 seconds
Memory limit: 512 megabytes

Tri is a boy who loves playing billiards. His wish is to become an international champion player like Bao Phuong Vinh, but because he did not have money to go to a billiards club to practice, he decided to make himself a billiards table to practice.

Unfortunately, **Tri**'s skills in making billiard tables are limited, so the billiard tables he creates are not as good as tables that meet international standards. Pointing out the disadvantages of the Billiards table that **Tri** created are countless, but the main disadvantages are:

- The billiard table may not be rectangular, but it is definitely a convex quadrilateral.

- Each time the ball collides with the edge of the table, the ball's velocity after colliding with the edge of the table is only half of the velocity before collision.

However, the billiard table that **Tri** created still satisfies the properties of a billiard table. When the ball collides with the edge of the table, it will reflect like light when it hits a flat mirror. Every point on the table surface has the same friction, meaning the ball always has a uniformly decreasing velocity when rolling on the table surface, called the backward acceleration $a\ cm/s^2$. To prove that the billiard table he made can still be used, **Tri** placed a red ball and a white circular marker on the table. The red ball and white marker both have a radius of r cm, their centers are respectively at coordinates (x_r, y_r) and (x_w, y_w). At first, the ball and the maker do not touch each other and are at least 1 cm away from the table wall. After that, **Tri** uses a billiard cue to create an initial velocity for the red ball so that when the red ball stops, it fits neatly at the marker position. Because **Tri** is not a professional billiards player, he cannot make the initial speed of the red ball greater than the value $v\ cm/s$. However, he wants to cover up the disadvantage as mentioned above, so he tries to hit the red ball as many times as possible on the edge of the table before touching the white maker. Knowing that, the ball and the maker are said to touch each other if the distance between their centers is less than or equal to the sum of their radii.

Question: How many positions can **Tri** create where the red ball collides with the edge of the table before reaching the white maker position?

Input

The first line is the integer coordinates of 4 consecutive adjacent vertices $(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4)$ with units of length is cm of the billiard table. Where, $1 \leq x_i, y_i \leq 1000$.

The second line contains 5 integers which are the integer coordinates of the center of the red ball (x_r, y_r), the coordinates of the center of the white marker (x_w, y_w), and their radius r (cm). The red ball and white marker are always satisfied to be inside the billiards table and at least 1 cm away from the table wall, and $1 \leq r \leq 3$.

The third line is two real numbers representing the backward acceleration a (cm/s^2) of the ball relative to the table surface and the initial velocity v (cm/s) of the red ball. Where, $1 < a \leq 1000$ and $1 \leq v \leq 1000$.

Output

The first line is an integer n representing the maximum number of positions where the red ball has collided with the table wall before reaching the position of the white marker without previously touching the maker. If the red ball cannot be brought to the white marker position, whether it touches the edge of the table at least once or not, print '0'.

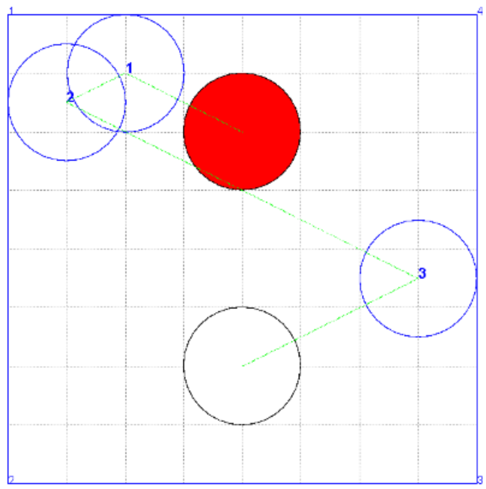
The next n lines, each line contains 2 real numbers x, y and 2 integers i, j respectively represent the coordinates of the center of the red ball (x, y) when it collides with the edge of the table, which is the straight line connecting i vertex and vertex j of the billiard table. These n lines are sorted in collision order. The maximum absolute error allowed is 10^{-6}. If there are multiple answers, you can print any one answer.

Examples

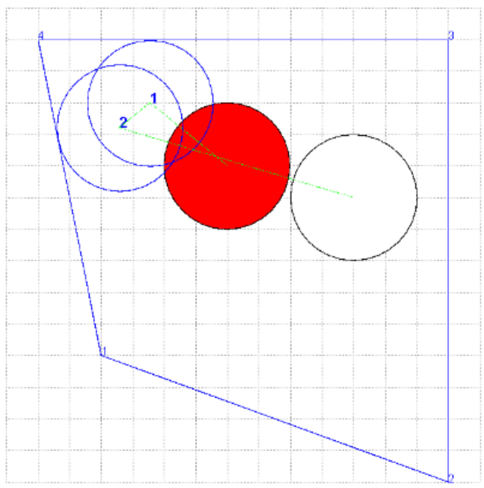
standard input	standard output
0 0 0 8 8 8 8 0 4 2 4 6 1 1.0 1000.0	3 2.0 1.0 4 1 1.0 1.5 1 2 7.0 4.5 3 4
3 11 14 15 14 1 1 1 7 5 11 6 2 2.0 500.0	2 4.554961773 3.0 3 4 3.596421722 3.784069582 4 1

Note

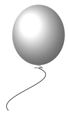
Description of input and output of example 1:



Description of input and output of example 2:



Problem L. Welcome

Ballon: 
Time limit: 1 seconds
Memory limit: 512 megabytes

Bi's task is to typeset the message printed on HUSC's electronic board for the ICPC Asian Hue 2023 contest. The message line is: "Welcome Hue University of Sciences". Instead of typesetting, **Bi** copies a multi-line text from the HUSC website with the purpose of choosing words to create the above message.

Ask the students to determine whether **Bi** received the message from the above text or not?

Input

Input consists of a number of lines, each with many words separated by one space characters.

Output

Print **Yes** if the above message appears and **No** otherwise (case sensitive).

Examples

standard input	standard output

`Welcome Hue`
`University`
`of Sciences`
	`Yes`

`welcome Hue University of Sciences`	`No`

`Welcome Hue city`
`Hue is the capital of Thua Thien`
`Hue province`
`University of Hue`
`of my life`
`Sciences scientific fields`
`or scientific disciplines`
	`Yes`

Problem M. Triangle in Triangle

Ballon:
Time limit: 1.5 seconds
Memory limit: 256 megabytes

> 10 talented individuals in combinatorics or dynamic programming cannot match the skills of one talented individual in geometry.
>
> Cao Cao did not say this quote

On the 2D Cartesian plane, you are given a non-degenerate triangle ABC. Your tasks are:

1. Construct the largest regular triangle PQR such that vertices P, Q, and R lie on segment AB, BC, and CA, respectively.
2. Construct the smallest regular triangle $P'Q'R'$ such that vertices P', Q', and R' lie on segment AB, BC, and CA, respectively.

Input

The first line contains the number of test cases t ($1 \leq t \leq 10^5$).

Each test case is described by a single line containing three pairs of integers (x_A, y_A), (x_B, y_B), (x_C, y_C) — the coordinate of the vertices of the given triangle ABC. The vertices are listed in counter-clockwise order.

The coordinates of all vertices are between -10^3 and 10^3. It is guaranteed that all triangles given in the input are non-degenerate.

Output

For each test case, print two real number — the edge length of the required largest and smallest triangle, respectively.

Definitions:

- p and P are the edge lengths of the smallest and largest triangles that you printed.
- j and J are the edge lengths of the smallest and largest triangles provided by the jury.

Your answer will be considered **correct** if the relative or absolute error between the edge length of the contestant's smallest and largest triangles to jury's is be no more than 10^{-9}, i.e., $\frac{|p-j|}{max(1,j)} \leq 10^{-9}$ and $\frac{|P-J|}{max(1,J)} \leq 10^{-9}$.

Example

standard input	standard output
4 0 0 1 0 0 1 4 4 1 1 10 1 4 6 1 1 10 1 4 4 1 1 5 1	0.732050807569 0.517638090205 3.105828541230 2.477529576886 5.718776704299 3.493284678235 3.105828541230 1.834583397065

Note

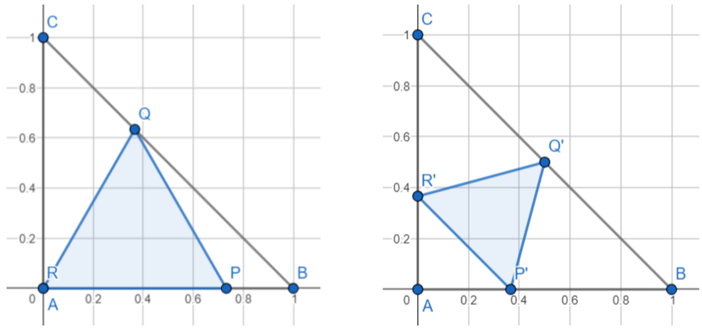


Illustration of the first sample.

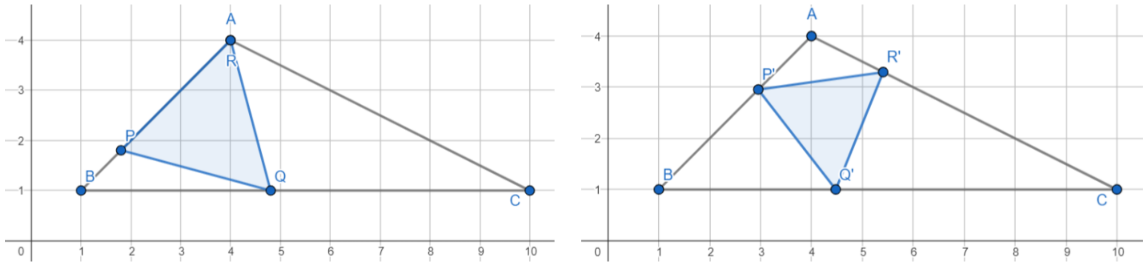


Illustration of the second sample.

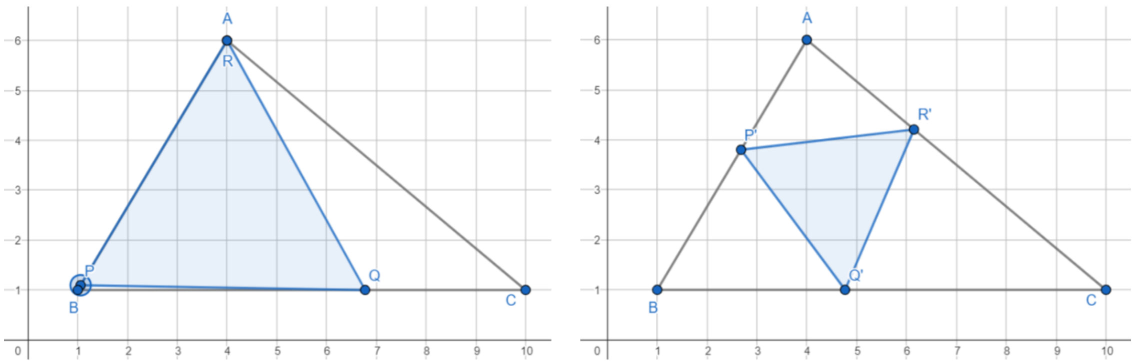


Illustration of the third sample.

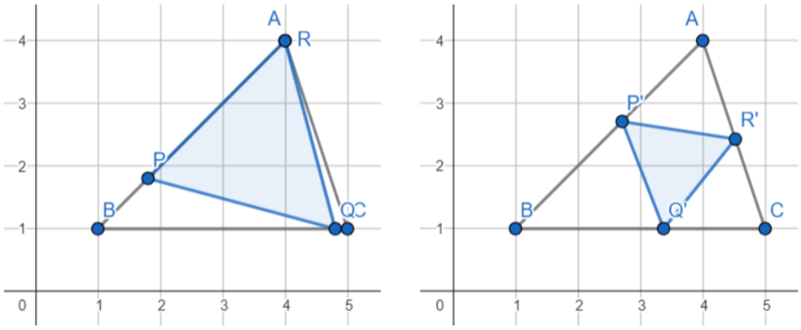


Illustration of the fourth sample.