

Problem A. Andrew and Efficient Change

Input file: **standard input**
 Output file: **standard output**
 Time limit: 1 second
 Memory limit: 256 megabytes

The kid Andrew doesn't like coins. His country currently uses n different coin types, each with a different value. It is guaranteed that the coin with value 1 is one of the different coin types.

Andrew's weekly taco-pizza grocery list contains $r - l + 1$ items with costs $l, l + 1, \dots, r$. Unfortunately, Andrew needs to buy each of these items separately as they are sold at different stores. Furthermore, these stores do not give any change, so Andrew must pay the exact amount in coins. Andrew is rather unhappy with the amount of coins he needs to handle every week, so he wants to convince his country to start using a new coin type to minimize the total number of coins needed to buy groceries for his weekly taco-pizza needs.

Andrew is too busy infiltrating UBC, so he wants you to tell him the best coin type to add, or that no new coin types can decrease the amount of coins he needs to use.

Input

The first line of the input will include an integer n ($1 \leq n \leq 420$), the number of coin types available.

The next line will include two integers l and r ($1 \leq l \leq r \leq 200000$, $r - l \leq 50$), the range of costs for Andrew's grocery list.

The next line will include n integers c_i ($1 \leq c_i \leq 200000$), the value of each coin type that is currently being used. It is guaranteed that no c_i is repeated, and there will be some i such that $c_i = 1$.

Output

The output should contain a single integer. If no new coin type can decrease the total number of coins used, print 0. Otherwise, print a single integer $1 \leq c \leq r$, the value of the coin type which, when added to the coin system, minimizes the total number of coins Andrew uses. If there are multiple solutions, output any.

Examples

standard input	standard output
1 10 10 1	10
3 10 15 1 5 10	12

Problem B. Paul's Badminton

Input file: **standard input**
 Output file: **standard output**
 Time limit: 5 seconds
 Memory limit: 256 megabytes

Paul is running a quaint business that delivers badminton rackets. There are n places of interest between which he will deliver rackets. These n places are connected by only $n - 1$ bidirectional roads because Paul's country does not believe in roads.

Paul has m employees. The i 'th employee has an order to deliver rackets from place a_i to place b_i , making one trip from place a_i to place b_i every day, from day s_i to day t_i inclusive.

Paul's country believes in tolls. Every day, if any of Paul's employees uses the j 'th road, Paul must pay an exorbitant toll of \$1 for that road. Fortunately, each toll must only be paid once per day. In other words, if Paul pays the toll for road j for some day, then any number of his employees may use that road during that day with no extra cost.

Paul wants to know some details on the amount of money he needs to pay for these tolls, so he gives you q queries. For the j th query, he asks you how much money he needs to pay from day c_j to day d_j inclusive. Send help.

Input

The first line of input contains two space-separated integers n , m , and q ($2 \leq n \leq 10^5$ and $1 \leq m, q \leq 10^5$), the number of places, employees, and queries respectively.

Each of the next $n - 1$ lines of input contains two space-separated integers x and y ($1 \leq x, y \leq n$), which means that there is a road connecting places x and y .

Each of the next m lines of input contains four space-separated integers a_i , b_i , s_i , and t_i ($1 \leq a_i, b_i \leq n$ and $1 \leq s_i \leq t_i \leq 10^9$), representing the order for the i 'th employee. It is guaranteed that $a_i \neq b_i$.

Each of the next q lines of input contains two space-separated integers c_j and d_j ($1 \leq c_j \leq d_j \leq 10^9$), representing Paul's j 'th query.

Output

The output should contain q lines. The j 'th line should contain a single integer which is the answer to the j 'th query (in the order given in the input).

Example

standard input	standard output
5 3 3	5
1 2	14
3 2	4
1 4	
1 5	
2 5 4 7	
3 4 2 5	
2 3 6 9	
7 11	
3 6	
5 5	

Problem C. Cyclic Song

Input file: **standard input**
 Output file: **standard output**
 Time limit: 5 seconds
 Memory limit: 256 megabytes

Eugene is inventing a new type of musical piece which has a few interesting properties.

A valid song of Type N has the following properties:

- The song only contains the notes A and B , all of equal length.
- The song will be repeated indefinitely. For example, if the song is AAB , then a musician will play $AABAABAABAABAAB\dots$ until Eugene falls asleep, which is once in a blue moon. This constitutes a *performance*.
- If we consider all the substrings of length N of the performance, the performance will contain **all 2^N possible strings of length N** .
- The representation of the song is **as short as possible**.

Here are some examples of valid and invalid songs:

- AB is a valid song of Type 1. It contains all substrings of length 1 (A and B) and it is as short as possible.
- $ABAB$ is **not** a valid song of Type 1, because a shorter representation (AB) exists.
- ABA is **not** a valid song of Type 2, because it is missing the substring BB .

Eugene would like you to create a *special* song of type N . He will specify two substrings S and T of length N . During a performance, the time between a performance of S and the next performance of T should be minimized. The performance of S and T can possibly overlap.

Formally, let X be the set of indices at which a substring equal to S begins, and let Y be the set of indices at which a substring equal to T begins. Then we want to minimize $\min\{y - x : x \in X \text{ and } y \in Y \text{ and } y \geq x\}$.

For example, for $N = 3$, $S = AAB$, and $T = ABA$, then $AABABBBBA$ makes Eugene happy (because AAB appears at position 1 and ABA appears at position 2, so the time between their performances is only 1), but $AABBBABA$ makes Eugene sad (because AAB appears at position 1, but ABA appears at position 6, so the time between their performances is 5, which is not minimal)

Eugene will be very happy if you can help him write a special song of Type N which minimizes the time from a performance of S to a performance of T .

Input

The first line contains an integer N ($2 \leq N \leq 20$), the type of the song.

The second line contains the string S , the first special substring that Eugene wants to see.

The third line contains the string T , the second special substring that Eugene wants to see.

It is guaranteed that $|S| = |T| = N$.

Output

Output a single line, a valid song of Type N that makes Eugene happy. If there are multiple such songs that make Eugene happy, output any of them. If Eugene cannot be happy, then output "SAD" (without quotation marks).

Examples

standard input	standard output
3 AAB ABA	AABABBBBA
3 ABA AAB	ABAAA BBBB

Problem D. David vs David

Input file: **standard input**
 Output file: **standard output**
 Time limit: 2 seconds
 Memory limit: 256 megabytes

David and David are playing a game. This game is played with two piles of stones, which we name pile 1 and pile 2. First, David and David decide on some non-negative number k which is called the tolerance. Then, David and David take turns making moves with David going first, and whoever cannot make a move loses. Each move is one of the following:

1. Remove any positive number of stones from one of the two piles
2. Remove a stones from pile 1 and b stones from pile 2, where $|a - b| \leq k$ and $a, b > 0$

David and David will play n games with different starting configurations. They want to check whether they played optimally, so for each game, they want you to tell them who should win with optimal play (and the answer is not `print('David')`).

Input

The first line of input contains a single integer n ($1 \leq n \leq 10^5$), representing the number of games that David and David will play.

Each of the next n lines contains three space-separated integers x , y , and k ($0 \leq x, y \leq 10^9$; $0 \leq k \leq 12$), representing a game with a tolerance k that starts with a pile of x stones and a pile of y stones.

Output

The output should contain n lines, each containing one integer. For each game, in the order given in the input, output a 1 if the first player wins, and a 2 if the second player wins.

Example

standard input	standard output
15	2
0 0 0	2
23 9 1	1
97 99 2	1
1984 6 3	1
277 348 4	1
2384 19 5	2
138 19 6	1
123 372 7	2
112 1021 8	1
99328 9702 9	1
3172 283401 10	2
1937 23405 11	1
421443 503539 12	2
508320368 822479633 0	1
924717293 228947159 1	

Problem E. Enegue's Enigmatic Lanterns

Input file: **standard input**
 Output file: **standard output**
 Time limit: 5 seconds
 Memory limit: 256 megabytes

Enegue is enjoying life as he strolls down a mildly overgrown path through the forest. At times, he could catch glimpses of the waning moon through gaps in the canopy. As the he followed the winding path through the undergrowth, Enegue suddenly felt the presence of a being somewhere just beyond his vision. Pausing momentarily, Enegue heard the rustling of leaves somewhere in the distance. Then, n lanterns appeared in a row in front of Enegue.

"Who," a voice asked.

"Enegue the eyqs," responded Enegue, "who are you?"

The voice simply replied "who." Then, with a bright flash, Enegue saw the silhouette of an owl before all the lanterns went dark. After a brief moment of darkness, k of the lanterns turned on again, and the owl was nowhere to be seen or heard. Suddenly struck by inspiration, Enegue decides to make you guess which k of the n lanterns are shining.

Enegue has a row of n lanterns, of which k are on. He wants you to guess which k of the n lanterns are on, but he will only answer your questions in a very cryptic fashion. For each question, you are allowed to choose any subset S of the n lanterns. Enegue will then count the number of lanterns on in S . Let this number be x , but Enegue will not tell you x . Instead, he will tell you the number of composite divisors of x . (eg. the composite divisors of 12 are $\{4, 6, 12\}$, so if $x = 12$, then Enegue will answer with 3).

Since Enegue is a cactus, he will get bored after 2×10^5 questions, so you'll need to guess the k lanterns that are on before he gets bored.

Input

The initial line of input contains two space-separated integers n and k ($4 \leq k \leq n \leq 100$), where n is the total number of lanterns, and k is the number of lanterns that are on.

Interaction Protocol

This is an interactive problem. Your program should use standard input and output to communicate with the judge's program.

You are allowed 2×10^5 queries. Each query should contain the character '?' and a string S of length n , separated by one space. The string S represents the chosen subset, where the i -th character of S is a '1' if and only if the i -th lantern is in the subset.

The response to each query will be a single integer as described above. A response of -1 means the query is malformed or you have submitted more than 2×10^5 queries, in which case your program should exit immediately and get a "Wrong Answer".

Once you are ready to guess the k lanterns, output a single line starting with the character '!' and a string T of length n , separated by one space. The string T represents the subset of k lanterns in the same format as the string S above. You will get "Accepted" if the guess is correct.

Example

standard input	standard output
9 5	// output to show interaction
0	? 111111111
0	? 000001111
	! 101100101

Note

The sample input and output only show how the interaction works. It will probably get wrong answer.

Problem F. Forever Young

Input file: `standard input`
Output file: `standard output`
Time limit: 1 second
Memory limit: 256 megabytes

Welcome to ACM Class! ACM Class has s students ($1 \leq s \leq 10^6$) of different ages, lined up along the wall to take attendance. The instructors Henry and Eugene each go from left to right and tick off everybody's name. They are bored of this menial task so they decide to play a game.

As Henry goes from left to right, he circles some students' names, following the rule that each student (after the first) is *older* than the last one he circled. Eugene circles some students' names so that each student is *younger* than the last one he circled. They are quite competitive, so using their **pro programming skillz** they each circle as many names as possible, as they know every student's age.

They will be very happy if Henry circles exactly n ($1 \leq n \leq s$) names and Eugene circles exactly m ($1 \leq m \leq s$), their respective favourite numbers. Furthermore, they are quite high achievers so cumulatively they are hoping to circle lots of names. Thus $n + m \geq s - 50$.

The class lines up differently every day. (Two lineups are different if at least one student is in a different position.) You'd like to keep Henry and Eugene happy for as long as possible to keep them from moving onto their next plan of starting World War 3. How long can you do this?

Input

The single line of input contains three space-separated integers s , n , and m ($1 \leq n, m \leq s \leq 10^6$). It is guaranteed that $n + m \geq s - 50$.

Output

Output the number of arrangements of the class that will make Henry and Eugene happy. This number can be quite large so output it modulo $10^9 + 7$.

Examples

standard input	standard output
6 3 3	256
12 3 4	213444

Problem G. Jonathan and Jason at the Jowling Jalley I

Input file: **standard input**
 Output file: **standard output**
 Time limit: 1 second
 Memory limit: 256 megabytes

Jonathan and Jason are huge weeaboos who spend all their time at home watching anime and reading light novels. But one day, the Google restaurant "Charlie's" was closed to all employees due to an invasion of wild UBC interns, so Jonathan and Jason spontaneously decided to go bowling. Unfortunately, the wild UBC interns got to the Google bowling alley "Gooowling Galley" first, and all the lanes were out of order, although actually they were out of order for several months already and the entire building was just a front for Baidu's secret invasion, but that's a problem for another time...

Basically Jonathan and Jason somehow ended up at a mysterious and sketchy location in Jupiter, the bowling alley "Jowling Jalley". The Martians are very skilled bowlers, so instead of using a mere 10 pins arranged in a triangle, with sides of length 4, the Martians use up to 210 pins arranged in a triangle, with sides of length $4 \leq n \leq 20$. Jonathan is very bad at bowling, so he forced Jason to play the "easy mode" variation with him, which is only popular among little children who watch cartoons.

In the "easy mode" variation, a hi-tech forcefield made out of op-amps and shafts under fatigue loading physically prevents the ball from knocking down any pins other than the first one, at the tip of the triangle. However, that first pin can still knock down the pins behind it, which can knock down more pins behind it, like a pile of Panago's. Basically, due to the hi-tech physics involved (it involves calculations of the von Mises stress using the Coulomb–Mohr criterion), the forcefield mysteriously warps causality in such a way that a pin can only be knocked down if both pins in front of it are also knocked down.

For example, after rolling the bowling ball, the first three positions below are possible, where o is an upright pin and x is a knocked-down pin. However, the last three positions are impossible, because at least one of the pins in front of pin X are not knocked down.

```

----o----  ---x---  ---x---  ----o----  ---x---  ---x---
---o_o---  ---x_o---  ---x_x---  ---X_o---  ---x_o---  ---x_o---
__o_o_o__  __x_o_o__  __x_x_o__  __o_o_o__  __o_X_o__  __x_o_o__
_o_o_o_o_  _o_o_o_o_  _x_o_o_o_  _o_o_o_o_  _o_o_o_o_  _x_X_o_o_
    
```

After every roll, a JeleVision (a telly made of jelly) displays a mysterious and sketchy animated video of a cartoon bowling ball knocking over cartoon bowling pins. This made Jonathan and Jason very excited because they are huge weeaboos. But they were also UBC students in the past, so naturally they wondered, how many animated videos did Jmazon (the company that owns the bowling alley "Jowling Jalley") have to produce? Jmazon has to produce exactly one animated video for every possible position that the pins can end up in, after Jonathan or Jason rolls the bowling ball once.

Input

The only line of the input is the integer n , the side length of the triangle of pins ($4 \leq n \leq 20$).

Output

Print the integer x , the number of possible positions after one roll in the Jowling Jalley, with the "easy mode" variation.

Example

standard input	standard output
4	42

Problem H. Quirky Queries

Input file: standard input
 Output file: standard output
 Time limit: 1.5 seconds
 Memory limit: 256 megabytes

Eugene, the number theorist, is studying some quirky integers. An integer x is quirky if there is no integer k such that k^2 is a factor of x , and there is no prime number $p \geq 300$ such that p divides x .

Eugene is playing with a sequence $a_1, a_2, \dots, a_n$ of quirky integers. Henry challenged Eugene to answer the following queries:

- Query of type 1 $l r x$: for all a_i in the range $[l, r]$, change a_i to x if the sorted sequence $S = \langle d_1, d_2, \dots, d_k \rangle$ of divisors of x is lexicographically smaller than the sorted sequence $T = \langle e_1, e_2, \dots, e_j \rangle$ of divisors of a_i . It is guaranteed that x is quirky.

For example, if Eugene had the integers 6, 10, 13 and the query 1 1 3 14, he would not change 6 or 10 since their sequences of divisors ($\langle 1, 2, 3, 6 \rangle$ and $\langle 1, 2, 5, 10 \rangle$ respectively) are lexicographically smaller than 14's sequence $\langle 1, 2, 7, 14 \rangle$, but he would change 13 to 14 since $\langle 1, 13 \rangle$ is lexicographically larger than $\langle 1, 2, 7, 14 \rangle$.

- Query of type 2 $l r$: output $\text{lcm}(a_l, a_{l+1}, \dots, a_r)$ modulo $10^9 + 7$.

Eugene needs to answer these quirky queries quickly! Can you help him?

Input

The first line contains a positive integer n ($1 \leq n \leq 10^5$).

The second line contains n integers $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq 10^7$). It is guaranteed that each a_i is quirky.

The third line contains a positive integer q ($1 \leq q \leq 2 \cdot 10^5$): the number of queries. Then follow q lines, each of the form 1 $l r x$ or 2 $l r$ ($1 \leq l \leq r \leq n$, $1 \leq x \leq 10^7$): queries of type 1 or 2, respectively. It is guaranteed that x is quirky.

Output

For each query of type 2 $l r$, output $\text{lcm}(a_l, a_{l+1}, \dots, a_r)$ modulo $10^9 + 7$.

Examples

standard input	standard output
3 6 10 13 5 1 1 3 14 2 1 1 2 2 2 2 3 3 2 1 3	6 10 14 210
1 6 5 2 1 1 1 1 1 2 2 1 1 1 1 1 1 2 1 1	6 2 1