Problem A. Find the metro station parent

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

After a long day at work, Yahya fell asleep on the metro. When he woke up, he found himself at an unfamiliar station and realized he had missed his stop! To get back on track, he needs to travel to the previous station (the parent station). However, in this peculiar metro system, the stations are organized in a unique tree structure.

In this metro network:

- Station 1 is the root station
- Each station number N has exactly N children
- Stations are numbered sequentially from 1
- Each station (except Station 1) has exactly one parent station

For example:

- Station 1 has one child: Station 2
- Station 2 has two children: Station 3 and 4
- Station 3 has three children: Station 5, 6, and 7
- Station 4 has four children: Station 8, 9, 10, and 11
- Station 5 has five children: Station 12, 13, 14, 15, and 16

Input

The input contains a single integer X $(2 \leq X \leq 10^9)$, representing the station where Yahya wakes up.

Output

Output a single integer representing the parent station number.

Examples

standard input	standard output
3	2
5	3

Problem B. Affordable Rent

Input file:	`standard input`
Output file:	`standard output`
Time limit:	6 seconds
Memory limit:	256 megabytes

A housing company, **StayEasy**, wants to implement a feature that allows users to query the most affordable rental property within a specific area.

Initially, a list of rental properties is registered on a 2D plane, where each property is uniquely positioned on the map with a fixed price. Following the registration phase, a series of queries will be made to determine the cheapest available rental within defined rectangular regions.

Each rental property is placed at a distinct location with a given price, and each search query specifies a bounded rectangular area within which to find the lowest-priced rental. The city is represented as a vast grid, where distances are measured using standard Cartesian coordinates.

Your task is to design a system that efficiently handles the registration of rental properties and then processes price-based search queries, ensuring users can quickly find the most affordable rental in their desired area.

Input

The first line contains an integer n, the total number of registered rentals ($1 \le n \le 2 \cdot 10^5$).

The next n lines describe the rental properties in the format:

name x y p : where name ($5 \le |\text{name}| \le 20$) is the unique name of the property, (x, y) are its coordinates ($0 \le x, y \le 10^9$), and p ($1 \le p \le 10^9$) is the price of the rental.

The next line contains an integer q, the number of queries ($1 \le q \le 2 \cdot 10^5$).

The next q lines describe the queries. Each query is of the form:

- x_1 y_1 x_2 y_2 : to find the cheapest rental property within the rectangular region bounded by (x_1, y_1) (bottom-left) and (x_2, y_2) (top-right), where x_1, y_1, x_2, y_2 are integers ($0 \le x_1, y_1, x_2, y_2 \le 10^9$) with $x_1 \le x_2$ and $y_1 \le y_2$.

Output

For each query, output a single line containing the name of the cheapest rental property within the specified region. If there are multiple properties with the same minimum price, print the one that was registered first. If no rental property is found in the region, output `"No property found in this region"`

Example

standard input	standard output
3 villa 10 10 150 cabin 5 8 90 loft 14 18 90 3 6 9 12 14 5 8 14 18 0 0 4 4	villa cabin No property found in this region

Note

The positions of the rental properties are distributed such that the variance of their coordinates is roughly equal in both the x and y directions.

Problem C. Minimum Operations

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

You are given two binary strings, S and T, both of length N ($1 \leq N \leq 10^5$). String T is guaranteed to have its most significant bit set to 1 and all the other bits set to 0.

You can perform the following operations repeatedly:

- If the leftmost bit of string S is set to zero, you can rotate string S one position to the left ('01011' $\rightarrow$ '10110').
- Increment the numerical value of string S by 1 ('1011' $\rightarrow$ '1100').

Your objective is to calculate the minimum number of operations required to transform S into string T using these two operations.

Input

The first line contains one integer t ($1 \leq t \leq 10^4$) — the number of test cases.

Each test case consists of two lines:

- The first line contains the string S ($1 \leq |S| \leq 2 * 10^5$), consisting of characters 0 or 1.
- The second line contains the string T ($|S| = |T|$), consisting of characters 0 or 1.

The sum of the lengths of strings $|S|$ and $|T|$ in all test cases does not exceed $2 * 10^5$.

Output

For each test case, output a single integer. Which is the minimum number of operations required to transform S into T, taken modulo $10^9 + 7$. If it is impossible to transform S into T, output -1.

Examples

standard input	standard output
1 00110110 10000000	11
1 1000111 1000000	-1

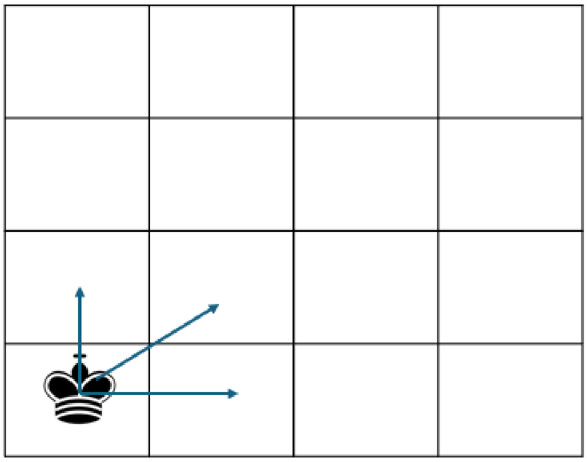
Problem D. Simple Game

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

Anas and Zakariaa are playing a game on an $m \times n$ grid. The grid has m rows and n columns, numbered from 1 to m (top to bottom) and 1 to n (left to right). A single piece (the king) starts at the bottom-left square, which is at position $(m, 1)$.

On each turn, the player in control must move the king to one of the following positions:

- One square to the right: $(r, c) \to (r, c + 1)$ (if $c + 1 \leq n$).
- One square up: $(r, c) \to (r - 1, c)$ (if $r - 1 \geq 1$).
- One square diagonally up-right: $(r, c) \to (r - 1, c + 1)$ (if $r - 1 \geq 1$ and $c + 1 \leq n$).



Anas always plays first. The two players take turns moving the king. The player who cannot make a move loses the game.

Given t ($1 \leq t \leq 100$) test cases, each with integers m and n ($1 \leq m, n \leq 10^6$), determine the winner if both players play optimally. Output `"Anas"` if Anas wins, and `"Zakariaa"` if Zakariaa wins.

Input

The first line contains a single integer t ($1 \leq t \leq 100$), the number of test cases.

Each of the next t lines contains two integers m and n ($1 \leq m, n \leq 10^6$), representing the grid dimensions.

Output

For each test case, output a single line containing either `"Anas"` or `"Zakariaa"` indicating the winner.

Example

standard input	standard output
1 2 2	Anas

Problem E. Defending Xyron

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

Omar is the strongest and fastest creature in Xyron (a land confined to the x-axis). The citizens of Xyron have elected him as their defender against enemy attacks. Fortunately, he has an excellent spy who provides him with the exact trajectories of incoming rockets before they reach Xyron. This allows Omar to intercept the rockets before they land.

Omar can move only along the x-axis, and he can intercept rockets only by sending defensive measures vertically (perpendicular to the x-axis). Your task is to help Omar determine the maximum distance between his position m_x (on the x-axis) and any point along the trajectory of the attacking rockets.

To achieve this, you will process Q queries of two types:

- Add a rocket trajectory given by the equation $y = ax + b$.
- Compute the maximum vertical distance between Omar's position and any point on the known rocket trajectories.

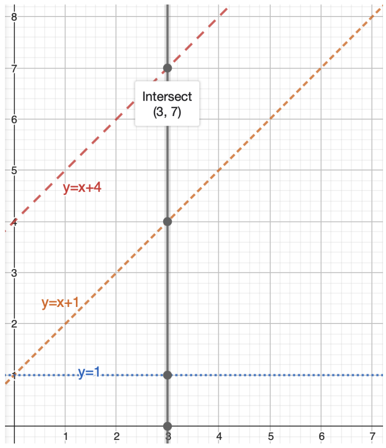


Illustration of the first test case.

Input

The first line of input contains an integer $2 \leq Q \leq 10^5$. The following Q lines contain queries in the form:

- `A` a b to add a rocket trajectory where $0 \leq a \leq 10^5$ and $0 \leq b \leq 10^5$;
- `G` m_x to get the maximum distance between Omar's position m_x and any point that the enemies' rockets will go through where $0 \leq m_x \leq 10^5$.

We note that a, b, and m_x are all integers. In addition, it is guaranteed that at least one `A` query will be before a `G` query and that there will not be any rockets falling in Xyron vertically.

Output

For each query of type `G`, output the maximum distance as described in the statement.

Examples

standard input	standard output
4 A 0 1 A 1 4 A 1 1 G 3	7
6 A 2 3 G 2 A 0 4 G 3 A 1 5 G 4	7 9 11

Problem F. Matrix Rank

Input file:	`standard input`
Output file:	`standard output`
Time limit:	4 seconds
Memory limit:	256 megabytes

Note: The definition of a rank of a matrix in this problem is different from the actual definition of the rank of a matrix in Algebra. Read the problem statement carefully!

The rank of a matrix in this problem is the minimum between (a) and (b) where

- (a) is the number of linearly independent column vectors in the matrix
- (b) is the number of linearly independent row vectors in the matrix.

The rank of the matrix cannot exceed the number of rows or columns.

Linear Independence: If a column/row of a matrix cannot be obtained by multiplying another column/row by an Integer scalar, then this column/row is linearly independent from any other column or row.

Example:

$$\begin{pmatrix} 3 & 6 & 3 \\ 2 & 3 & 4 \\ 1 & 2 & 1 \end{pmatrix}$$

The rank of the matrix above is 2 because multiplying the third row $[1, 2, 1]$ by 3 results in $[3, 6, 3]$ which is equal to the first row. This means that we have one row linearly dependent on another (thus a is equal to 2). There is no Linear dependence between columns (b is equal to 3).

Input

The first line of the input contains two numbers $1 \leq N \leq 500$ and $1 \leq M \leq 500$ denoting respectively the number of rows, and the number of columns of the matrix. The matrix is defined in the following lines. N rows follows, each containing M numbers $1 \leq a_{ij} \leq 10^9$.

Output

Output the rank of the matrix as described int the statement.

Examples

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

Problem G. A Multiplication Problem

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

You are with your favorite math professor, explaining how hard it is to do multiplication in your head. Luckily, you are smart enough to write a program to automate this for you.

Your task is simple, write a program that verifies if the multiplication of two given numbers $A * B$ is larger or equal to C.

Input

First and only line contains 3 integers $0 \leq A, B, C \leq 10^{18}$.

Output

Print Yes if the multiplication of A and B is larger or equal to C and No otherwise.

Examples

standard input	standard output
10 10 100	Yes
6 7 43	No

Problem H. Numerical Journey

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

You are given an array of n **integers**. Each number in the array can **"jump"** to another number if they are **not coprime**, meaning the greatest common divisor (GCD) of the two numbers is greater than 1.

Your task is to determine whether **every number in the array is reachable from every other number** using any number of jumps.

Input

- The first line contains a single integer n $(1 \leq n \leq 10^5)$ — the number of elements in the array.
- The second line contains n space-separated integers $a_1, a_2, \ldots, a_n$ $(1 \leq a_i \leq 10^6)$.

Output

Print `"yes"` if it is possible to reach any number in the array from any other number using any number of jumps. Otherwise, print `"no"`.

Examples

standard input	standard output
4 2 3 6 12	yes
4 2 5 10 7	no

Note

In the second example, it's impossible to reach 7 starting from any other number.