The International Collegiate Programming Contest
Sponsored by ICPC Foundation

The 2026 ITI Placement Test Collegiate Programming Contest

(Contest Problems)

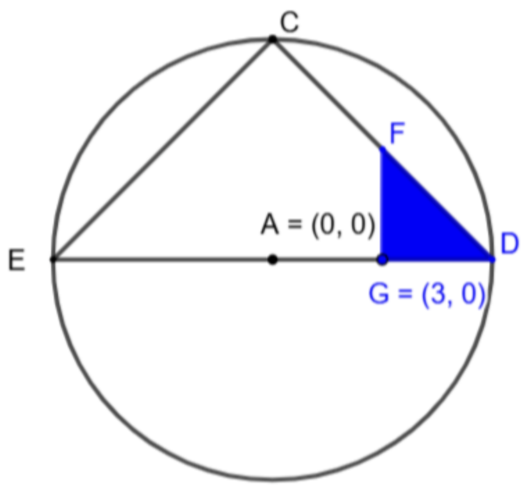
Online
February 2026

Problem A. Inscribed triangle in circle

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Teal`

Two points A and G lie in the plane. Point A is the center of a circle, and point G lies strictly inside the circle.

You are given the area of triangle $\triangle ECD$, which is a right triangle inscribed in the circle such that points E, C, and D lie on the circle, and segment ED is a diameter that passes through both points A and G.



Let point F be the foot of the perpendicular dropped from point G onto segment CD.

Determine the area of triangle $\triangle FGD$.

Input

The first line contains an integer t $(1 \le t \le 10^5)$ — the number of test cases.

The first line of each test case contains two integers x_a, y_a — coordinates of point A $(-10^5 \le x_a, y_a \le 10^5)$.

The second line contains two integers x_g, y_g — coordinates of point G. It's guaranteed that $G \ne A$ and lies strictly inside the circle.

The third line contains one integer S $(4 \le S \le 10^6)$ — the area of triangle $\triangle ECD$.

Output

For each test case, print the area of triangle $\triangle FGD$.

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

Example

standard input	standard output
3 0 0 3 0 36 28 14 30 13 100 -62 -61 -93 -83 2209	4.500000 30.139320 40.381686

Problem B. Max Equal Subarray

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Teal`

You are given an array a of length n consisting of non-negative integers.

You may perform the following operation any number of times:

- Choose any position i such that $a_i = 0$, and change it to any non-negative integer.

After performing any number of such operations, determine the maximum possible length of a **contiguous subarray** consisting of the same value.

Input

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

Each test case starts with an integer n $(1 \le n \le 10^5)$ — the number of elements in the array.

The second line contains n integers $a_1, a_2, \ldots, a_n$ $(0 \le a_i \le 10^9)$.

It is guaranteed that the total number of elements over all test cases does not exceed 10^5.

Output

For each test case, print a single integer — the length of the longest subarray consisting of the same value after any number of allowed operations.

Example

standard input	standard output
3 9 2 4 0 3 0 0 3 0 4 5 1 1 5 2 2 4 0 0 0 0	6 2 4

Problem C. Saving for a Birthday Gift

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Teal`

Ahmed is Amr's brother, and Amr's birthday is in exactly X days. Starting today, Ahmed can save exactly Y dinars per day until the birthday. He wants to buy a gift that costs Z dinars.

Determine whether Ahmed can save enough money to buy the gift by Amr's birthday.

Input

A single line contains three integers X, Y, and Z ($1 \le X, Y, Z \le 100$) — the number of days until the birthday, the amount Ahmed saves each day, and the price of the gift.

Output

Print `YES` if Ahmed can buy the gift by the birthday; otherwise, print `NO`.

Examples

standard input	standard output
`4 2 3`	`YES`
`4 9 40`	`NO`

Problem D. Same-Color Pairs

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Teal`

You have N items, each colored with a lowercase English letter. You want to form as many **pairs** as possible, where each pair must consist of two items of the **same** color. Each item can be used in at most one pair.

Compute the maximum number of such pairs.

Input

The first line contains a single integer N ($1 \leq N \leq 2 \cdot 10^5$).

The second line contains a string S of length N consisting of lowercase English letters, where S_i is the color of the i-th item.

Output

Print a single integer — the maximum number of valid (same-color) pairs you can form.

Examples

standard input	standard output
4 abca	1
7 abacbab	2

Problem E. Game

Input file: `standard input`
Output file: `standard output`
Balloon Color: `Teal`

Alaa and Habiba played a game where Alaa scored A and Habiba scored H.

Can you determine who won or if it was a draw?

Input

One line contains two integers number A, H $(1 \le A, H \le 100)$.

Output

Print "A"if Alaa won, "H"if Habiba won, or "D"if it was a draw.

Examples

standard input	standard output
2 3	H
3 1	A

Problem F. Domty Distribution

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Teal`

In the ECPC competition, each team consists of 3 members. The organizers have a certain number of **Domty** and want to distribute them evenly among the teams. You are given the number of **Domty** available, and your task is to determine if it can be evenly distributed among the three members of each team

Input

A single integer $d(1 \le d \le 10^9)$, which represents the total number of Domty available.

Output

Print "YES" if the Domty can be evenly distributed among the three members of each team, otherwise print "NO".

Examples

standard input	standard output
3333	YES
3334	NO

Problem G. AI-Enhanced Password

Input file: standard input
Output file: standard output
Balloon Color: Teal

Hamed is working on a secret account, but the platform's manager has just rolled out a new AI security model that enforces dynamic password policies. As Hamed types his new password, the AI model interrupts with a cryptic directive:

the password must contain its length.

Hamed found it very hard and asked you to solve this problem, he wants his password to stay the same, but you can add digits $[0-9]$ to the end of the password to represent a number (without leading zeros), so at the end that number is the length of the password. can you do it?

Input

the only input contains one string S $(1 \le |S| \le 100)$ representing Hamed's original password, it is guaranteed that the last character is not a digit.

Output

the only output contains one string , the new password that contains its length as a suffix.

Examples

standard input	standard output
123456a	123456a8
hello	hello6
aaabbbccc	aaabbbccc11

Problem H. Legacy Systems Validation

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Teal`

In the not-so-distant future, the world is powered by intelligent AI agents. Each year, the Global AI Qualification Challenge is held to identify and celebrate the most brilliant AI minds from across the globe. This challenge pushes them to their limits—solving puzzles, learning at breakneck speeds, and even simulating ethical decision-making.

Those who reach the final round are honored with a special commemorative badge, a symbol of excellence and ingenuity. The badge proudly displays the year of the agent's achievement. But there's a catch...

The organization in charge of printing the badges relies on a vintage badge-printing machine, a relic from the early 21st century. This old hardware is finicky and fragile—it crashes if asked to print any year that contains more than 3 distinct digits. The engineers say updating the system would "ruin its historical charm"(and also cost too much).

As the head of badge validation, your task is to write a program to determine whether a given 4-digit year is safe to be printed using this ancient machine.

Input

One integer Y — the year ($1000 \leq Y \leq 9999$).

Output

Print **"YES"** if the badge can be printed, otherwise print **"NO"**.

Examples

standard input	standard output
1923	NO
2004	YES

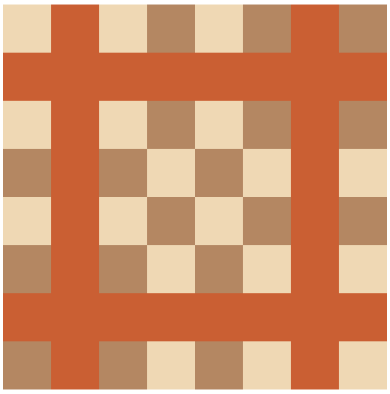
Problem I. Eman's Board

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Teal`

In the third week, a small bug appeared. No one could find it. They gave the code to the AI, but it didn't help—it didn't understand code they didn't write. They forgot how to debug by themselves.

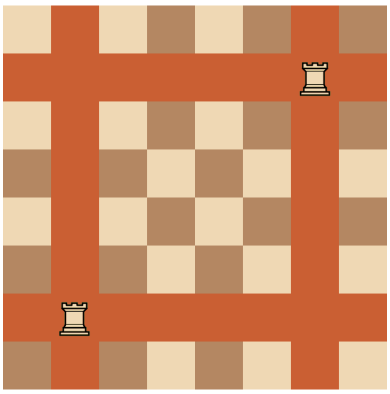
Eman has an $N \times M$ chessboard, where N represents the number of rows and M represents the number of columns. Rocks will be placed on the board such that **all marked squares must be attacked**, and **no unmarked square can be attacked**.

She now seeks to determine the maximum and minimum number of rocks that can be placed on the board to produce the same marked squares. For example, consider the following initial board with marked attack squares:

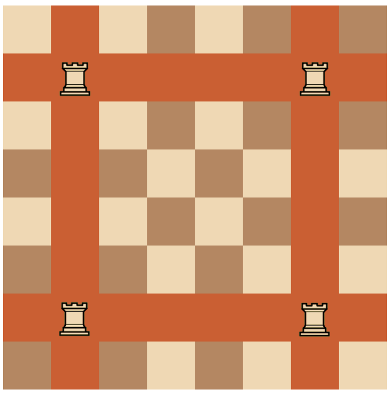


Initial board with marked attack squares

For this board, the maximum number of rocks that can be placed to recreate the same marked squares is 4, while the minimum number is 2. The configurations are shown below:



Configuration with 2 rocks



Configuration with 4 rocks

Input

The first line contains two integers N and M ($1 \leq N, M \leq 10^3$), denoting the dimensions of the chessboard.

The following N lines contain M characters each, representing the initial state of the board. Each character can be either a '#' representing a marked square or a '.' representing an unmarked square.

It is guaranteed that the board is valid — that is, there exists at least one configuration of rocks that can produce the given pattern of marked squares.

Output

Output two integers, separated by a space, representing the maximum and minimum number of rocks that can be placed on the board to make the same marks.

Examples

standard input	standard output
8 8 .#....#. ######## .#....#. .#....#. .#....#. .#....#. ######## .#....#.	4 2
2 2 .# ##	1 1

Problem J. Towering Triumph

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Teal`

The team had just discovered a revolutionary AI coding assistant. With a single sentence, it could write entire functions. At first, it felt like magic—an endless stream of working solutions. But none of the team members could explain how the solution worked. Still, they moved on.

During the ECPC Qualification, the team's balloon got stuck on the ceiling! The only way to retrieve it is by stacking their three chairs, and the tallest of the three teammates has to make the climb.

Given their three heights, find the height of the climber.

Input

The first and only line of input contains three space-separated integers: h_1, h_2, h_3 $(150 \leq h_1, h_2, h_3 \leq 210)$ – representing the heights of the three teammates in centimeters.

Output

Print a single integer, which is the height of the tallest teammate.

Example

standard input	standard output
`155 160 180`	`180`

Problem K. Boring Lecture

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Teal`

The AI often gave similar ideas: greedy, then DP, then brute force. The team stopped thinking of new ways. Their creativity became weaker. They just waited for the same old ideas.

In a boring lecture held in a hall represented as a two-dimensional grid of size $N \times M$, each cell contains a seat.

The person sitting in seat (i, j) will get bored after $A_{i,j}$ minutes from the start of the lecture and will want to leave.

However, the person will only leave if two conditions are satisfied:

- At least $B_{i,j}$ people have already left the lecture before them.
- There exists a path from their seat to any edge of the grid such that all seats along the path are empty (i.e., already vacated), allowing the person to walk out.

Once both conditions are satisfied, the person leaves immediately, without consuming any additional time.

Your task is to determine, for each person, at what time he will leave the lecture, or if he will never leave.

Input

The first line contains two integers N and M ($1 \le N \times M \le 3 \times 10^5$) — the number of rows and columns of the grid.

Then N lines follow, each containing M integers $A_{i,j}$ ($0 \le A_{i,j} \le 10^9$) — the minute each person starts wanting to leave.

Then N lines follow, each containing M integers $B_{i,j}$ ($0 \le B_{i,j} \le N \times M$) — the minimum number of people that must have left before the person at (i, j) is willing to go.

Output

Print N lines, each containing M integers. The value at position (i, j) should represent the minute at which the person in that seat leaves the lecture.

If a person never leaves, print `-1` instead.

Examples

standard input	standard output
2 2 1 0 3 4 1 2 0 1	3 3 3 4
4 4 5 0 9 2 9 4 15 7 9 2 7 12 0 15 14 15 12 6 2 16 11 1 1 1 6 5 16 6 0 0 4 2	15 15 9 -1 15 15 15 7 15 15 -1 15 0 15 15 15

Problem L. The Factorial Manuscript of Elbasan

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Teal`

Elbasan, a fortified city that flourished under Ottoman rule, became a regional center for Islamic scholarship in the Balkans. Within its libraries, mathematicians and scribes studied patterns of numbers, factorials, and modular arithmetic. One preserved manuscript posed a timeless challenge:

Given a number n, compute the value of $0! + 1! + 2! + \cdots + n!$.

But even centuries ago, scholars knew that factorials grew too quickly to handle directly. The manuscript thus included a clever restriction: all answers must be computed modulo $10^7 + 7$.

Now, your task is to carry on the legacy of Elbasan's mathematicians and solve this query efficiently, even for very large values of n.

Input

The first line contains an integer t ($1 \le t \le 10^5$) — the number of test cases.

Each of the following t lines contains a single integer n ($0 \le n \le 10^{18}$).

Output

For each test case, print the value of $0! + 1! + \cdots + n!$ modulo $10^7 + 7$.

Example

standard input	standard output
2 1 3	2 10

Problem M. Palindromes of Hama

Input file: `standard input`
Output file: `standard output`
Balloon Color: `Teal`

Hama, a historic city on the banks of the Orontes River in Syria, was a vibrant center of engineering and mathematics during the Islamic Golden Age. Its giant norias (water wheels) rotated in perfect symmetry — inspiring scholars to study patterns, cycles, and mirrored structures. One preserved manuscript from Hama describes a curious number puzzle involving transformations and palindromes.

You are given a permutation a of size n (that is, $a = [1, 2, \ldots, n]$ in some order). Initially, you have 0 coins.

You may perform the following operations any number of times, in any order, as long as your coin balance never becomes negative:

- **Decrement Operation**: Decrease any element of the array by 1 and gain 1 coin.
- **Increment Operation**: Spend 1 coin to increase any element of the array by 1.

You may freely modify the array using these operations, but all final values in the array must remain **non negative integers**.

Your goal is to maximize the number of **distinct palindromic numbers** in the final array.

A palindromic number is a positive integer whose decimal representation reads the same forwards and backwards (e.g., 1, 22, 131).

Input

The first line contains an integer t ($1 \leq t \leq 10^6$) — the number of test cases.

Each of the next t lines contains a single integer n ($1 \leq n \leq 10^9$) — the size of the permutation.

Output

For each test case, print one integer — the maximum number of distinct palindromic numbers you can form in the array after performing the allowed operations.

Example

standard input	standard output
2 9 20	9 14

Problem N. The Records of Jerusalem

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Teal`

Jerusalem — known as Bayt al-Maqdis in Islamic tradition — is a city of immense historical and spiritual importance. From the Umayyad to the Ottoman periods, scholars and merchants passed through its gates, often tracking values like trade weights or commodity prices. In one ancient manuscript, a mathematical puzzle describes measuring the density of values within specific numeric windows.

You are given an array a of size n, and an integer len.

Define the function $f(L, R)$ as the maximum length of a subarray where every element is between L and R (inclusive). That is, $f(L, R)$ is the maximum number of consecutive elements in a such that every value in that segment satisfies $L \le a_i \le R$.

Your task is to find the maximum possible value of $f(L, R)$ over all valid pairs (L, R) such that $R - L = len$.

Note:

- The length of a value range $[L, R]$ is $R - L$ (not $R - L + 1$).
- The length of a subarray $[l, r]$ is $r - l + 1$.

Input

The first line contains a single integer t ($1 \le t \le 10^5$) — the number of test cases.

The first line of each testcase contains two integers n and len ($1 \le n \le 10^6$, $1 \le len \le 10^9$) — the length of the array and the value range length.

The second line of each testcase contains n integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 10^9$) — the elements of the array.

It is guaranteed that the sum of all n across all test cases does not exceed 10^6.

Output

For each test case, print one integer — the maximum value of $f(L, R)$ over all valid ranges $[L, R]$ with $R - L = len$.

Example

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

Problem O. Even Digits in Latakia

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Teal`

While wandering the harbor streets of Latakia, Abady noticed something peculiar. On old shipping labels and merchant tokens, some numbers just looked... luckier. He made up a little rule to amuse himself: if both digits of a two-digit number looked 'calm' and 'balanced', he smiled. Otherwise, he moved on.

Given a two-digit number N, help Abady decide whether to smile or not. He smiles only if both digits are even.

Input

Only one line conatins single integer N ($10 \leq N \leq 99$).

Output

Print `YES` if both digits of N are even, otherwise print `NO`.

Examples

standard input	standard output
24	YES
35	NO

Problem P. Great XOR

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Teal`

John and Mariam are best friends who love solving puzzles together. One evening, Mariam comes up with a bitwise challenge to test John's sharpness.

She gives him a single integer x and says:

Find how many positive integers a are there such that:

- $0 < a < x$
- $(a \oplus x) > x$

where $\oplus$ means the **bitwise XOR** operation.

John thinks it sounds easy, but as he starts writing out binary representations, he realizes the problem hides a clever trick. Now he needs your help to solve it!

Input

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

Each of the next t lines contains one integer x ($2 \leq x \leq 10^{10}$).

Output

For each test case, output the answer for the given value of x.

Example

standard input	standard output
2 2 5	1 2

Problem Q. MTI or MIT

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Teal`

In the bustling world of academia, two renowned universities, MTI and MIT, engage in friendly competition and academic excellence. Yet, a peculiar phenomenon surrounds their names ... MTI and MIT are not just universities but strings of infinite possibilities.

Consider the strings:

1. *mti*, which repeats as *mtimtimtimti*... infinitely.

2. *mit*, echoing as *mitmitmitmit*... infinitely.

Your task is to transform a given string S into a substring of either *mtimtimtimti*... or *mitmitmitmit*... with the least number of operations. Each operation allows you to change any character in S as needed.

However, there's a twist! If the minimum operations to fit S into either symphony are equal, output **"FAKE"** to signify the impossibility of differentiation.

Input

The first line contains an integer N ($1 \le N \le 2 \times 10^5$), the length of the string S.

The second line contains the string S of length N.

Output

Print **"FAKE"** if the minimum number of operations required to make the string S a substring of both patterns is equal, Otherwise, print "*mti X*" if the minimum number of operations is smaller for the pattern "*mti*", or "*mit X*" if it is smaller for the pattern "*mit*", where X is the minimum number of operations.

Examples

standard input	standard output
`9` `mtimitabc`	`FAKE`
`7` `mtmierw`	`mit 4`

Problem R. gcd ?

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Teal`

$Maryam$ is studying *gcd* for the first time, and she thought it was pretty cool. So she decided to give you a simple challenge just for fun!

She gives you an array a of n integers and q queries. Each query consists of a single integer k.

For each query, determine whether it is possible to **change at most one element of the array** a **(i.e., change zero or one elements)**

such that the greatest common divisor of the resulting array gcd$(a_1, a_2, a_3, ..., a_n)$ **becomes exactly equal to** k **.**

Input

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

The second line contains n **integers** $a_1, a_2, a_3, ..., a_n$ $(1 \leq a_i \leq 10^7)$ **.**

The third line contains q $(1 \leq q \leq 10^6)$ **queries , each query contains an integer** k $(1 \leq k \leq 10^9)$ **.**

Output

For each query output "Yes" if it is possible, and "No" otherwise.

You can output the answer in any case (upper or lower). For example, the strings "yEs" , "yes", "Yes", and "YES" will be recognized as positive responses.

Example

standard input	standard output
4 4 16 8 32 4 1 2 3 4	YES YES NO YES

Problem S. I Hate Minimums

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Teal`

You are given an array a of n integers and q queries. Each query is defined by two integers l and r, representing a segment of the array.

For each query, your task is to count the number of **subarrays** $[a_i, a_{i+1}, \ldots, a_j]$ such that:

- $l \le i \le j \le r$ (i.e., the subarray is fully contained within the interval $[l, r]$), and
- The **minimum value** of the selected subarray is **equal** to the minimum value of the elements **outside** the subarray but **within** the query range $[l, r]$.

In other words, you are to compare the minimum of a subarray inside $[l, r]$ with the minimum of the *remaining elements* in $[l, r]$ that are not in that subarray.

Note: If the subarray covers the entire range $[l, r]$, then the remaining elements are empty, and their minimum is considered to be 0.

Input

The input consists of:

- A single integer n $(1 \le n \le 10^5)$ — the size of the array a.
- An array a of n positive integers, where each element satisfies $1 \le a[i] \le 10^9$.
- A single integer q $(1 \le q \le 10^5)$ — the number of queries.
- q queries, each in the form of two integers l, r $(1 \le l \le r \le n)$.

Output

output answer for each query

Example

standard input	standard output
9 1 3 2 2 2 2 2 3 2 5 6 8 5 7 2 3 9 9 4 6	3 5 0 0 5

Problem T. Unstable Draw Game

Input file: `standard input`
Output file: `standard output`
Balloon Color: `Teal`

Anne and Brian are playing a game with a deck of N cards. The rules are as follows:

- Anne always plays first.
- On each turn, a player must remove **1**, **2**, or **3** cards from the deck.
- A player must not remove the **same number of cards** as the previous player.
- If the **absolute difference** between the number of cards removed by the current player and the previous player is **2** (a prime number), the current player **immediately loses**.
- The player who removes the last card wins, as long as no rules are violated.
- **On the first move only**, Anne can choose any valid move (**1**, **2**, or **3**) without restrictions. That is, Rules 3 and 4 do not apply to the first move.

Both players play optimally. Determine who will win the game.

Input

The first line contains a single integer T ($1 \le T \le 10^5$) — the number of test cases.

Each test case contains one integer N ($1 \le N \le 10^8$) — the initial number of cards in the deck.

Output

For each test case, print a single line containing the name of the winner: `Anne` or `Brian`.

Assume that both players play optimally.

Example

standard input	standard output
1 7	Anne

Note

Both players play optimally.

A player immediately loses if their move violates Rule 4.

The game always ends with one winner.

Problem U. Welcome to school

Input file: `standard input`
Output file: `standard output`
Balloon Color: `Teal`

Every morning, a teacher and a student come to class. You are given the hours when the student arrives and the teacher arrives (both expressed as integers on a 12-hour clock).

If the student arrives strictly before the teacher, the student says "YES"; otherwise, the student says "NO".

Input

The input consists of two lines.

The first line contains a single integer a ($1 \le a \le 12$) — the hour when the student arrives.

The second line contains a single integer b ($1 \le b \le 12$) — the hour when the teacher arrives.

Output

Output

Print **"YES"** if the student arrives strictly before the teacher, and **"NO"** otherwise.

Examples

standard input	standard output
3 9	YES
5 5	NO

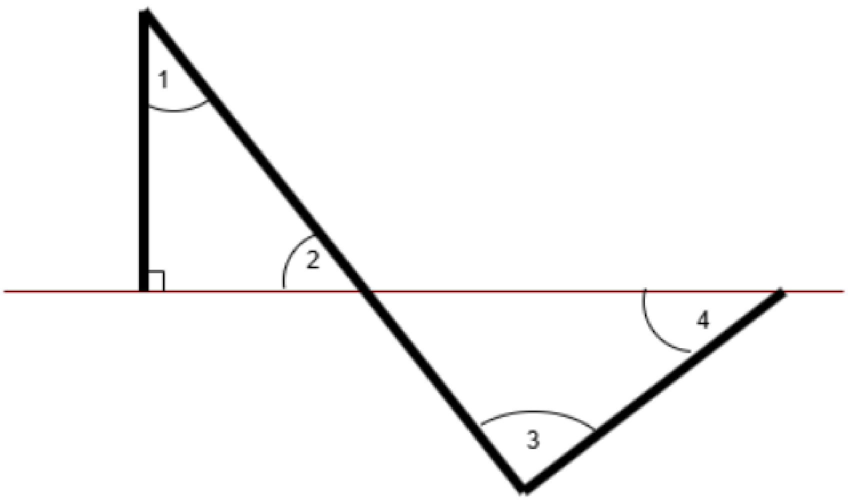
Problem V. Geo!Geo!Geo!!!

Input file: standard input
Output file: standard output
Balloon Color: Teal

In a geometric shape with four angles 1, 2, 3, and 4, you are given the values of two of them:

One of the angles among 1 or 2, and one of the angles among 3 or 4.

Determine the values of all the remaining angles, or state that it is impossible to form a valid shape with the given angles.



Input

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

Each test case consists of two lines:

The first line contains two integers i and a ($i \in \{1,2\},\ 0 \le a \le 90$) — the index and value of the first known angle.

The second line contains two integers j and b ($j \in \{3,4\},\ 0 \le b \le 180$) — the index and value of the second known angle.

Output

For each test case, print four integers — the values of angles 1, 2, 3, and 4 in this exact order.

If it's impossible to determine valid angles for the shape, print `-1`.

Example

standard input	standard output
2 2 30 4 90 1 90 3 60	60 30 60 90 -1

Problem W. Homz and the Laser

Input file: `standard input`
Output file: `standard output`
Balloon Color: `Teal`

Homz loves puzzles — especially ones with lasers and mirrors. One day, he stumbled upon a strange $H \times W$ grid carved into the floor of an old lab. Each tile of the grid had either nothing, or a shiny mirror: some were slash-shaped (`/`), others backslash-shaped (`\`). Curious and excited, Homz decided to play.

He places a laser at an **entry point** along the edge of the grid, aiming it in one of four directions: `U` (up), `D` (down), `L` (left), or `R` (right). Then, the fun begins!

As the laser travels through the maze:

- If it hits an empty cell (`.`), it just keeps going in the same direction.
- If it hits a mirror, it reflects based on the mirror type:
  - A `"/"` mirror bends the laser:
    * `U` → `R`, `R` → `U`, `D` → `L`, `L` → `D`
  - A `"\"` mirror bends it differently:
    * `U` → `L`, `L` → `U`, `D` → `R`, `R` → `D`

The laser continues to travel and reflect until it eventually escapes the grid. Homz marks a target **exit point** on the boundary, along with the direction he wants the laser to exit in.

But wait! Some mirrors might be placed wrongly... Luckily, Homz is allowed to toggle up to K mirrors — flipping a `"/"` into a `"\"` and vice versa.

laser travel from cell to another in one second, you can flip the mirror for one second then it will return back to its original direction.

Your job is to help Homz determine the **the minimum number of flips required** (no more than K), so that the laser can travel from the entry to the exit in the desired direction. If it's impossible, even with K flips, print `-1`.

Can you guide Homz's laser safely through the maze?

Input

The first line contains three integers H, W, and K ($1 \le H, W \le 1000$, $0 \le K \le 10^6$) — the height and width of the grid, and the maximum number of toggles allowed.

The second line contains three entries: integers r_s, c_s (1-based row and column of the entry point), and a character dir_s representing the initial direction (`U`, `D`, `L`, or `R`).

The third line contains three entries: integers r_e, c_e (the exit point), and a character dir_e (the required exit direction).

Then H lines follow, each containing a string of W characters — the grid itself, made up of characters `"."`, `"/"`, or `"\"`.

Output

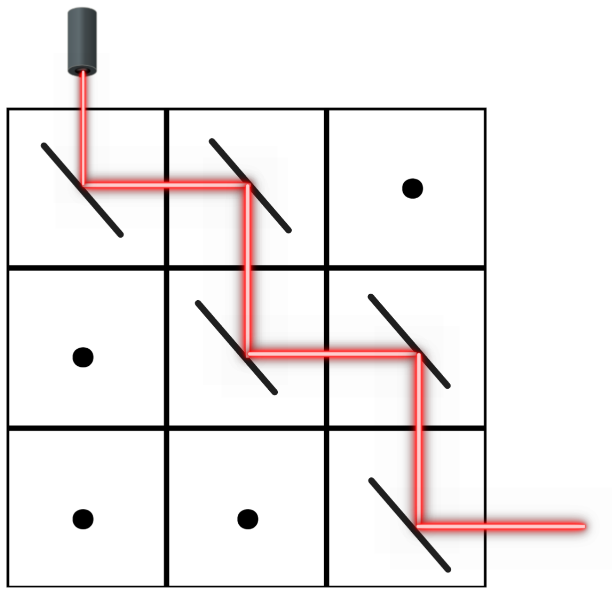
Print a single integer — the minimum number of flips required to make the laser reach the desired exit cell and direction. If it is impossible within K toggles, print -1.

Examples

standard input	standard output
3 3 1 1 1 D 3 3 D .\. /.\ ...	1
3 3 0 1 1 D 3 3 R \\. .\\ ..\	0
4 5 2 1 3 D 4 1 R ..... ./.\. ..../	-1

Note

for the second testcase the laser can exit from the exit point with no changes of the mirrors as illustrated here:



Problem X. Digit Hunt

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Teal`

"Welcome to the Squid Code. 14 rounds.
Only coders survive"

The masked judge

A masked guard drops a pile of digits at your feet. "You must be able to display every number from* 1 *to* N *— one at a time. Choose your cards wisely. You get no spares."

You are given an integer N.

You want to buy digit cards (from 0 to 9) so that you can form every number from 1 to N (inclusive), using the cards to represent one number at a time.

What is the minimum total number of digit cards you need to buy?

Input

The only line contains a single integer N ($1 \le N \le 10^{18}$).

Output

Print the minimum number of digit cards needed.

Example

standard input	standard output
`11`	`11`

Note

To write all numbers from 1 to 11, you need the following digit cards: $0, 1, 1, 2, 3, 4, 5, 6, 7, 8, 9$ — that is, 11 cards in total.

Problem Y. Count Primeful Subtrees

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Teal`

You are given a tree with n vertices, rooted at vertex 1. Each vertex i has a positive integer value a_i.

The **subtree of a vertex** v is the set consisting of v and all of its descendants in the rooted tree. Thus, there are exactly n subtrees, one for each vertex.

The **value** of a subtree is defined as the product of the values a_i over all vertices belonging to that subtree.

A subtree is called **primeful** if the number of **distinct prime numbers** in the prime factorization of its value is **odd**.

Your task is to count the number of such primeful subtrees.

Input

The first line contains a single integer n $(1 \le n \le 2 \cdot 10^5)$ — the number of vertices in the tree.

The second line contains n integers $a_1, a_2, \ldots, a_n$ $(1 \le a_i \le 10^6)$ — the value of each vertex.

Each of the next $n - 1$ lines contains two integers u and v $(1 \le u, v \le n,\ u \ne v)$ — an edge between vertices u and v.

It is guaranteed that the input describes a tree.

Output

Print a single integer — the number of primeful subtrees.

Example

standard input	standard output
3 2 3 5 1 2 1 3	3

Note

There are three possible subtrees rooted at the three vertices:

- Subtree rooted at vertex 1 includes all nodes: $2 \times 3 \times 5 = 30$, whose prime factors are $\{2, 3, 5\}$ — 3 distinct primes (odd).
- Subtree rooted at vertex 2: value is 3 — 1 prime (odd).
- Subtree rooted at vertex 3: value is 5 — 1 prime (odd).

All have an odd number of distinct prime factors, so the answer is 3.

Problem Z. Folka and Movies

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Teal`

Folka downloaded some series of movies of total n movies but they aren't sorted so it is hard for Folka to watch them in order.

The movie name is guaranteed to be in this form `id-name`. For example `1-ironman`,`012-spongebob`

Folka is a bit messy, as he moved all movies in one folder so now the folder contains all series more simplified, for example the folder contains spiderman-series, ironman-series and spongebob-series and others.

So, Folka wants to watch them in the smallest lexicographically order (of movies name) and for each series he wants to watch the series in order of its indexing (episode1 before episode3).

Can you tell him the order he should follow?

Since Folka is using a program to read you output and start playing the movie, print the name as it is in the input so Folka's program won't crash. i.e, don't remove or add or change any character in name.

Input

The first line contains a single integer $n(1 \le n \le 5 \cdot 10^5)$, the number of movies.

The next n lines, on the i_{th} line contains a single string s_i which is the naming of the i_{th} movie.

It is guaranteed that the movie naming :

- is in this form `id-name`
- no name has two distinct id
- the name contains at least one lowercase english alphabet letters and may be integer at its end without leading zeores
- the length of s_i doesn't exceed 19

For example `1-ironman`,`012-spongebob`

Output

print the order of movies as Folka wants.

Examples

standard input	standard output
5 1-ironman1 2-ironman2 1-ironman2 2-ironman1 5-antman	5-antman 1-ironman1 2-ironman1 1-ironman2 2-ironman2
2 010-afolka 011-zfolka	010-afolka 011-zfolka