Problem A. Alice

TimeLimit: 1 second
MemoryLimit: 256 megabytes

Alice and Bob play a game on a tree with n nodes, initially all colored white. They take turns performing the following operation:

- Choose a white node x and color all nodes on the path from node x to node 1 black. If some nodes on the path are already black, this doesn't affect the operation (they simply remain black).

Alice goes first. If a player has no valid move on their turn, they lose. Determine who will win the game assuming both play optimally.

A **tree** with n nodes is a connected graph with exactly $n-1$ edges, and there exists a unique simple path between any pair of nodes.

Input

The first line contains an integer n ($1 \leq n \leq 2 \times 10^5$) — the number of nodes. Each of the next $n-1$ lines contains two integers u and v ($1 \leq u, v \leq n$), representing an edge between nodes u and v.

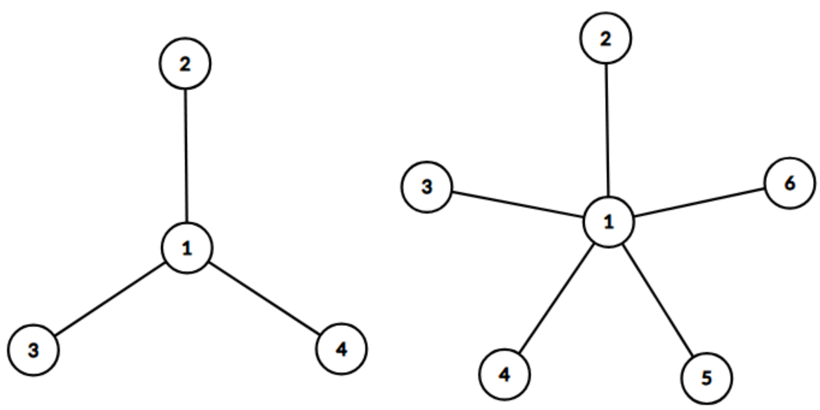
Output

Print "`Alice`" if Alice will win the game, or "`Bob`" if Bob will win the game.

Examples

standard input	standard output
4 1 2 1 3 1 4	Alice
6 5 1 1 3 2 1 4 1 1 6	Alice

Note



The tree of example 1 and 2.

Problem B. Daily Schedule

TimeLimit:	1 second
MemoryLimit:	256 megabytes

Since the ICPC season has arrived, Warner and his teammates want to practice more and build a regular routine so that they can perform well in the official contest. Therefore, they decided to practice one contest every day.

Warner's daily routine is defined as follows:

- If the time is in the interval `[10:30, 15:30)`, Warner is in **practice** programming contest team training.
- If the time is in the interval `[07:00, 23:00)` but not during practice, Warner is **awake**.
- Otherwise, Warner is **asleep**.

All intervals are left-closed and right-open (inclusive at the start, exclusive at the end). The practice interval is a sub-interval of the awake interval.

Input

The input consists of two integers, `hh` and `mm`, separated by a space, representing the time in 24-hour format.

- $0 \leq \texttt{hh} \leq 23$
- $0 \leq \texttt{mm} \leq 59$

Output

Given a time, print `PRACTICE` if Warner is in practing, `AWAKE` if he is awake but not practicing, and `ASLEEP` otherwise.

Examples

standard input	standard output
14 30	PRACTICE
23 59	ASLEEP
7 0	AWAKE
23 0	ASLEEP

Problem C. Median Consensus

TimeLimit:	1 second
MemoryLimit:	256 megabytes

A renowned statistician is studying a new method for measuring the consensus of a dataset. The method involves analyzing the "representative value" of every possible focus group that can be formed from the data.

The statistician starts with a dataset containing n numerical measurements $A_1, A_2, \ldots, A_n$. A "focus group" is simply any **non-empty subset** of these n measurements. For any given focus group, its "representative value" is defined as its **median**.

The statistician uses a specific definition for the median: for a group with k elements, the median is the element that would be at the $\lfloor \frac{k+1}{2} \rfloor$-th position if the group were sorted in non-decreasing order. For example:

- The median of $\{10, 50, 20\}$ is the 2-nd smallest element, which is 20.
- The median of $\{7, 1, 9, 3\}$ is the 2-nd smallest element, which is 3.

To find the overall "Total Consensus Value" of the entire dataset, the statistician wants to sum the representative values of all $2^n - 1$ possible non-empty focus groups.

Given the initial n measurements, can you calculate the Total Consensus Value? Since this sum can be extraordinarily large, you must compute the result modulo 998 244 353.

Input

The first line of input contains a single integer n ($1 \leq n \leq 2 \times 10^5$), the number of measurements in the dataset.

The second line contains n space-separated integers $A_1, A_2, \ldots, A_n$ ($0 \leq A_i \leq 10^9$), representing the measurements.

Output

Print a single integer: the Total Consensus Value modulo 998 244 353.

Examples

standard input	standard output
3 3 3 2	18
4 2 2 1 4	28
6 0 2 3 1 2 3	106
8 6 1 7 0 4 3 3 6	853

Problem D. A 01 Problem

TimeLimit:	1 second
MemoryLimit:	256 megabytes

OIL is a freshman majoring in Computer Science. In his introductory course to computer science, he is currently learning about binary conversion. However, OIL often confuses some characters:

- The digit `0` and the letter `O` are considered the same.
- The digit `1` and the letter `l` are considered the same.

Given a string s consisting only of the characters `0`, `1`, `l`, `O`, OIL wants to avoid making mistakes, OIL decides to first replace all English letters with their corresponding digits. Then, he wants to compress the string s.

String compression means merging consecutive identical characters into a single character. The operation is applied until no two adjacent characters in the string are the same.

Your task is to output the resulting string after his operation.

Input

The input consists of two lines. The first line contains an integer n ($1 \le n \le 2 \times 10^5$), the length of the string. The second line contains a string s of length n, consisting only of the characters `0`, `1`, `l`, and `O`.

Output

The result string.

Examples

standard input	standard output
8 Ol0O1lOO	01010
10 0l01O0OO1l	010101

Problem E. Scam Hideouts

TimeLimit: 3 seconds
MemoryLimit: 256 megabytes

A large scam syndicate operates across a wide territory. For logistical planning, they represent all hideout locations as coordinates on a 2D plane. In each of the n cities where they operate, they have two available hideouts: a primary one and a backup.

To stay flexible, they must choose **exactly one** hideout to use from the two options in each city.

The leader of the syndicate wants all the chosen hideouts to be as close to each other as possible. If the hideouts are too far apart, it becomes difficult to move people or equipment in an emergency. Therefore, they want to make a selection that makes the "maximum distance between any two chosen hideouts" as small as possible.

For their calculations, they always use the **squared Euclidean distance** to avoid complex square root operations.

Your task is to find the minimum possible value of this "maximum squared distance".

Input

The first line of input contains a single integer n ($1 \le n \le 1000$), the number of cities.

The next n lines each describe the two hideouts in a city. Each line contains four integers: x_1, y_1, x_2, y_2, representing the coordinates of the two hideouts, (x_1, y_1) and (x_2, y_2). The coordinate values are between -10^9 and 10^9.

It is guaranteed that all $2n$ locations are distinct.

Output

Print a single integer: the minimum possible value of the maximum squared distance.

Examples

standard input	standard output
1 -8 -76 -75 0	0
3 1 1 6 2 -1 1 -2 5 0 3 1 7	5
2 -5 1 5 1 1 3 -1 5	20

Problem F. Houyi's Ultimate Orchard

TimeLimit: 1 second
MemoryLimit: 256 megabytes

Legend has it that Houyi shot down the suns not to save the world, but to create a ultimate orchard for growing the most delicious fruit. He shot down n suns and placed them in orbit around his garden. These suns orbit the garden in a cycle of m time units.

Generated by Google Gemini.

At time j, the i-th sun contributes a_{ij} to the garden's temperature. The total temperature of the garden at time j is the sum of the contributions from all the suns, $\sum_{i=1}^{n} a_{ij}$.

To achieve the perfect flavor, Houyi discovered the secret lies in maximizing the temperature difference. He can use his divine powers to independently adjust the orbit of each sun, cyclically shifting its temperature contribution over time. For example, if a sun's temperature contribution sequence is $[t_1, t_2, \cdots, t_m]$, he can shift it to $[t_2, \cdots, t_m, t_1]$ or any other cyclic permutation.

Your task is to help Houyi find the optimal sun orbit configuration that maximizes the difference between the highest and lowest temperatures of the day. This maximum temperature difference is defined as the "deliciousness" of the fruit.

Input

The first line contains two integers n and m — the number of suns and the time cycle, respectively.

The following n lines each contain m integers, where the j-th number on the i-th line, a_{ij} , represents the temperature contribution of the i-th sun at time j.

- $1 \leq n, m \leq 300$
- $0 \leq a_{ij} \leq 9$

Output

Output a single integer representing the "deliciousness" of the fruit, which is the maximum possible temperature difference.

Examples

standard input	standard output
2 3 1 1 4 5 1 4	7
10 10 1 8 2 5 6 5 5 0 8 3 4 9 6 0 9 8 9 6 6 4 1 7 4 2 3 2 2 2 2 2 5 9 3 8 9 0 3 8 8 1 9 3 3 8 9 9 0 6 0 1 2 2 8 4 3 7 7 2 3 7 6 3 3 1 7 4 8 0 4 0 5 8 2 9 5 9 1 8 8 6 3 4 7 7 1 4 6 9 2 7 5 6 9 1 0 4 4 9 5 9	69

Note

In the sample input 1, by shifting the first sun's sequence to `[4, 1, 1]`, we achieve a maximum temperature of 9 (from $4 + 5$) and a minimum of 2 (from $1 + 1$), resulting in a final difference of 7.

Problem G. Collecting NCU

TimeLimit: 1 second
MemoryLimit: 256 megabytes

At the National Coding University (NCU), students discovered a strange message string S. The department chair announced a special rule: whenever students collect one subsequence “NCU” from a given string, they can skip class for one day.

Formally, you are given a string S consisting only of the characters `N`, `C`, and `U`.

Each collected subsequence “NCU” must take its characters in order from S, and each position in S can be used in at most one subsequence.

Your task is to calculate the maximum number of “NCU” subsequences that can be collected from S.

A subsequence is a sequence that can be derived from a given string by deleting zero or more elements, while maintaining the relative order of the remaining elements. For example, "ace" and "abd" are both subsequences of "abcde".

Generated by Google Gemini.

Input

The first line contains an integer T $(1 \leq T \leq 2 \times 10^5)$ — the number of test cases.

Each test case consists of two lines. The first line contains an integer n $(1 \leq n \leq 2 \times 10^5)$, the length of the string. The second line contains a string S of length n, which consists only of the characters `N`, `C`, and `U`.

It is guaranteed that the total length of all strings does not exceed 2×10^5.

Output

For each test case, output a single integer – the maximum number of “NCU” subsequences that can be collected from the corresponding string.

Example

standard input	standard output
3 9 NCNUCUNCU 6 NCNUUC 6 UCNUCN	3 1 0

Problem H. Medusa

TimeLimit: 1 second
MemoryLimit: 256 megabytes

Generated by Google Gemini.

In a timeless cavern, n Medusas are arranged in a circle, each lost in a deep slumber. The wake-up schedule for each Medusa follows a precise, repeating pattern.

The set of scheduled wake-up times for the i-th Medusa, which we can denote as W_i, forms an infinite arithmetic progression. This progression is defined by an initial term t_i and a common difference v_i. The set of times can be formally expressed as:

$$W_i = \{t_i + k \cdot v_i \mid k \in \mathbb{Z}_{\geq 0}\}$$

This means Medusa i is scheduled to wake up at times $t_i, t_i + v_i, t_i + 2v_i, t_i + 3v_i$, and so on.

A powerful curse governs their slumber. At any moment in time T, if the set of Medusas scheduled to wake up contains **more than one member**, a **petrification event** occurs. All Medusas in this set—that is, all who wake up at time T—will see each other and are instantly turned to stone.

Once a Medusa is petrified, she is permanently removed from the simulation. Her future scheduled wake-ups are canceled and will not occur. These events are resolved chronologically. The group of Medusas involved in the earliest petrification event is removed first, before any subsequent events are considered.

Your task is to analyze these schedules and determine the fate of each Medusa. For each of the n Medusas, find the exact time at which she turns to stone. If a Medusa will never be part of a petrification event, she is considered to sleep safely forever.

Input

The first line of input contains a single integer n ($1 \leq n \leq 1000$), the number of Medusas.

The following n lines each describe a Medusa's schedule. The i-th of these lines contains two integers, t_i and v_i ($0 \leq t_i \leq 10^6$, $1 \leq v_i \leq 10^6$).

Output

Print a single line containing n space-separated integers. The i-th integer should be the time at which the i-th Medusa turns to stone. If the i-th Medusa will never turn to stone, print `-1` instead.

Examples

standard input	standard output
3 3 2 0 2 5 9	5 -1 5
3 0 1 3 9 3 4	3 3 3
4 0 4 1 4 2 4 3 4	-1 -1 -1 -1
5 0 11 2 5 3 6 4 7 5 11	11 27 27 11 27

Problem I. Collecting NCU 2

TimeLimit: 1 second
MemoryLimit: 256 megabytes

Welcome, future NCU legends, to the university's most prestigious event: The Grand Sticker Challenge!

Your mission, should you choose to accept it, is to collect three special stickers: `N`, `C`, and `U`. For every complete set of these three stickers you collect, you'll earn a well-deserved day off from classes!

There are n students participating in this year's hunt. The university has planned q exciting events to distribute the stickers. In each event, a specific type of sticker will be given to a continuous group of students, from student l to student r.

After all q events are finished, your task is to determine the total number of days off each student has earned.

Generated by Google Gemini.

Input

The first line contains two integers, n and q.

The following q lines each contain three values: two integers l and r, and a character c.

- $1 \le n, q \le 2 \times 10^5$
- $1 \le l \le r \le n$
- c is an uppercase letter.

Output

Output a single line containing n integers. The i-th integer should represent the number of days off for the i-th student.

Examples

standard input	standard output
5 7 1 5 N 5 5 C 3 4 P 1 3 C 2 2 I 4 4 L 2 5 U	0 1 1 0 1
1 7 1 1 N 1 1 N 1 1 N 1 1 C 1 1 C 1 1 U 1 1 U	2
3 7 1 1 N 1 1 N 1 1 N 2 2 C 2 2 C 3 3 U 3 3 U	0 0 0
3 6 1 1 W 1 2 A 1 3 R 2 2 N 2 3 E 3 3 R	0 0 0