A. Odd One Out

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

You are given three digits a, b, c. Two of them are equal, but the third one is different from the other two.

Find the value that occurs exactly once.

Input

The first line contains a single integer t ($1 \le t \le 270$) — the number of test cases.

The only line of each test case contains three digits a, b, c ($0 \le a, b, c \le 9$). Two of the digits are equal, but the third one is different from the other two.

Output

For each test case, output the value that occurs exactly once.

Standard Input	Standard Output
10 1 2 2 4 3 4 5 5 6 7 8 8 9 0 9 3 6 3 2 8 2 5 7 7 7 7 5 5 7 5	1 3 6 7 0 6 8 5 5 7

,

B. Not Quite Latin Square

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

A *Latin square* is a 3×3 grid made up of the letters A, B, and C such that:

- in each row, the letters A, B, and C each appear once, and
- in each column, the letters A, B, and C each appear once.

For example, one possible Latin square is shown below.

$$\begin{bmatrix} A & B & C \\ C & A & B \\ B & C & A \end{bmatrix}$$

You are given a Latin square, but one of the letters was replaced with a question mark ?. Find the letter that was replaced.

Input

The first line of the input contains a single integer t ($1 \leq t \leq 108$) — the number of testcases.

Each test case contains three lines, each consisting of three characters, representing the Latin square. Each character is one of A, B, C, or ?.

Each test case is a Latin square with exactly one of the letters replaced with a question mark ?.

Output

For each test case, output the letter that was replaced.

Standard Input	Standard Output
3 ABC C?B BCA BCA CA? ABC ?AB BCA ABC	A B C

Note

The correct Latin squares for the three test cases are shown below:

$$\begin{bmatrix} A & B & C \\ C & \mathbf{A} & B \\ B & C & A \end{bmatrix} \quad \begin{bmatrix} B & C & A \\ C & A & \mathbf{B} \\ A & B & C \end{bmatrix} \quad \begin{bmatrix} \mathbf{C} & A & B \\ B & C & A \\ A & B & C \end{bmatrix}$$

,

C. Can I Square?

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

Calin has n buckets, the i-th of which contains a_i wooden squares of side length 1.

Can Calin build a square using **all** the given squares?

Input

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

The first line of each test case contains a single integer n ($1 \leq n \leq 2 \cdot 10^5$) — the number of buckets.

The second line of each test case contains n integers $a_1, \ldots, a_n$ ($1 \leq a_i \leq 10^9$) — the number of squares in each bucket.

The sum of n over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, output "YES" if Calin can build a square using **all** of the given 1×1 squares, and "NO" otherwise.

You can output the answer in any case (for example, the strings "yEs", "yes", "Yes" and "YES" will be recognized as a positive answer).

Standard Input	Standard Output
5 1 9 2 14 2 7 1 2 3 4 5 6 7 6 1 3 5 7 9 11 4 2 2 2 2	YES YES NO YES NO

Note

In the first test case, Calin can build a 3×3 square.

In the second test case, Calin can build a 4×4 square.

In the third test case, Calin cannot build a square using all the given squares.

,

D. Unnatural Language Processing

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

Lura was bored and decided to make a simple language using the five letters `a`, `b`, `c`, `d`, `e`. There are two types of letters:

- *vowels* — the letters `a` and `e`. They are represented by `V`.
- *consonants* — the letters `b`, `c`, and `d`. They are represented by `C`.

There are two types of *syllables* in the language: `CV` (consonant followed by vowel) or `CVC` (vowel with consonant before and after). For example, `ba`, `ced`, `bab` are syllables, but `aa`, `eda`, `baba` are not.
A *word* in the language is a sequence of syllables. Lura has written a word in the language, but she doesn't know how to split it into syllables. Help her break the word into syllables.

For example, given the word `bacedbab`, it would be split into syllables as `ba.ced.bab` (the dot `.` represents a syllable boundary).

Input

The input consists of multiple test cases. The first line contains an integer t ($1 \le t \le 100$) — the number of test cases. The description of the test cases follows.

The first line of each test case contains an integer n ($1 \le n \le 2 \cdot 10^5$) — the length of the word.

The second line of each test case contains a string consisting of n lowercase Latin characters — the word.

All words given are valid words in the language; that is, they only use the letters `a`, `b`, `c`, `d`, `e`, and each word is made up of several syllables.

The sum of n over all test cases does not exceed $2 \cdot 10^5$.

Output

For test case, output a string denoting the word split into syllables by inserting a dot `.` between every pair of adjacent syllables.

If there are multiple possible splittings, output any of them. The input is given in such a way that at least one possible splitting exists.

Standard Input	Standard Output
6 8 bacedbab 4 baba 13 daddecabeddad 3 dac 6	ba.ced.bab ba.ba dad.de.ca.bed.dad dac dac.dac da.bab.ba.ba.bab.bab.ba.bab.ba

dacdac 22 dababbababababbababbba	

,

E. Romantic Glasses

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

Iulia has n glasses arranged in a line. The i-th glass has a_i units of juice in it. Iulia drinks only from odd-numbered glasses, while her date drinks only from even-numbered glasses.

To impress her date, Iulia wants to find a contiguous subarray of these glasses such that both Iulia and her date will have the same amount of juice in total if only the glasses in this subarray are considered. Please help her to do that.

More formally, find out if there exists two indices l, r such that $1 \leq l \leq r \leq n$, and $a_l + a_{l+2} + a_{l+4} + \cdots + a_r = a_{l+1} + a_{l+3} + \cdots + a_{r-1}$ if l and r have the same parity and $a_l + a_{l+2} + a_{l+4} + \cdots + a_{r-1} = a_{l+1} + a_{l+3} + \cdots + a_r$ otherwise.

Input

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

The first line of each test case contains a single integer n ($1 \leq n \leq 2 \cdot 10^5$) — the total number of glasses.

The second line of each test case contains n integers $a_1, \ldots, a_n$ ($1 \leq a_i \leq 10^9$) — the amount of juice in each glass.

The sum of n over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, output "YES" if there exists a subarray satisfying the condition, and "NO" otherwise.

You can output the answer in any case (for example, the strings "yEs", "yes", "Yes" and "YES" will be recognized as a positive answer).

Standard Input	Standard Output
6 3 1 3 2 6 1 1 1 1 1 1 10 1 6 9 8 55 3 14 2 7 2 8 1 2 11 4 1 5 1 2 6 2 6 1 5 7 8 9 2 5 10 4 4 9 6 7 8	YES YES NO YES NO YES

Note

In the first test case, Iulia can pick $l = 1$ and $r = 3$. Then she drinks $a_1 + a_3 = 1 + 2 = 3$ units and her date drinks $a_2 = 3$ units of juice.

In the second test case, Iulia can pick $l = 2$ and $r = 5$. Then she drinks $a_3 + a_5 = 1 + 1 = 2$ units and her date drinks $a_2 + a_4 = 1 + 1 = 2$ units of juice.

In the third test case no such contiguous subarray works.

In the fourth test case, Iulia can pick $l = 2$ and $r = 8$. Then she drinks $a_3 + a_5 + a_7 = 11 + 1 + 1 = 13$ units and her date drinks $a_2 + a_4 + a_6 + a_8 = 2 + 4 + 5 + 2 = 13$ units of juice.

,

F. Greetings

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

There are n people on the number line; the i-th person is at point a_i and wants to go to point b_i. For each person, $a_i < b_i$, and the starting and ending points of all people are distinct. (That is, all of the $2n$ numbers $a_1, a_2, \ldots, a_n, b_1, b_2, \ldots, b_n$ are distinct.)

All the people will start moving simultaneously at a speed of 1 unit per second until they reach their final point b_i. When two people meet at the same point, they will greet each other once. How many greetings will there be?

Note that a person can still greet other people even if they have reached their final point.

Input

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

The first line of each test case contains a single integer n ($1 \le n \le 2 \cdot 10^5$) — the number of people.

Then n lines follow, the i-th of which contains two integers a_i and b_i ($-10^9 \le a_i < b_i \le 10^9$) — the starting and ending positions of each person.

For each test case, all of the $2n$ numbers $a_1, a_2, \ldots, a_n, b_1, b_2, \ldots, b_n$ are distinct.

The sum of n over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, output a single integer denoting the number of greetings that will happen.

Standard Input	Standard Output
5 2 2 3 1 4 6 2 6 3 9 4 5 1 8 7 10 -2 100 4 -10 10 -5 5 -12 12 -13 13 5 -4 9	1 9 6 4 0

-2 5 3 4 6 7 8 10 4 1 2 3 4 5 6 7 8	

Note

In the first test case, the two people will meet at point 3 and greet each other.

,

G. Bicycles

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

All of Slavic's friends are planning to travel from the place where they live to a party using their bikes. And they all have a bike except Slavic. There are n cities through which they can travel. They all live in the city 1 and want to go to the party located in the city n. The map of cities can be seen as an undirected graph with n nodes and m edges. Edge i connects cities u_i and v_i and has a length of w_i.

Slavic doesn't have a bike, but what he has is money. Every city has exactly one bike for sale. The bike in the i-th city has a slowness factor of s_i. Once Slavic buys a bike, he can use it **whenever** to travel from the city he is currently in to any neighboring city, by taking $w_i \cdot s_j$ time, considering he is traversing edge i using a bike j he owns.

Slavic can buy as many bikes as he wants as money isn't a problem for him. Since Slavic hates traveling by bike, he wants to get from his place to the party in the shortest amount of time possible. And, since his informatics skills are quite rusty, he asks you for help.

What's the shortest amount of time required for Slavic to travel from city 1 to city n? Slavic can't travel without a bike. It is guaranteed that it is possible for Slavic to travel from city 1 to any other city.

Input

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

The first line of each test case contains two space-separated integers n and m ($2 \leq n \leq 1000$; $n - 1 \leq m \leq 1000$) — the number of cities and the number of roads, respectively.

The i-th of the following m lines each contain three integers u_i, v_i, w_i ($1 \leq u_i, v_i \leq n$, $u_i \neq v_i$; $1 \leq w_i \leq 10^5$), denoting that there is a road between cities u_i and v_i of length w_i. The same pair of cities can be connected by more than one road.

The next line contains n integers $s_1, \ldots, s_n$ ($1 \leq s_i \leq 1000$) — the slowness factor of each bike.

The sum of n over all test cases does not exceed 1000, and the sum of m over all test cases does not exceed 1000.

Additional constraint on the input: it is possible to travel from city 1 to any other city.

Output

For each test case, output a single integer denoting the shortest amount of time Slavic can reach city n starting from city 1.

Standard Input	Standard Output
3 5 5 1 2 2 3 2 1 2 4 5	19 36 14

2 5 7 4 5 1 5 2 1 3 3 5 10 1 2 5 1 3 5 1 4 4 1 5 8 2 3 6 2 4 3 2 5 2 3 4 1 3 5 8 4 5 2 7 2 8 4 1 7 10 3 2 8 2 1 4 2 5 7 2 6 4 7 1 2 4 3 5 6 4 2 6 7 1 6 7 4 4 5 9 7 6 5 4 3 2 1	