

Problem A. Arrow Array

Input file: **standard input**
 Output file: **standard output**
 Time limit: 1 second
 Memory limit: 1024 megabytes

Mr. Nežmah has an array of length n and he wants to fill it up with arrows pointing either to the left or to the right. However, he has $n - 1$ conditions he would like to be satisfied. For each i from 1 to $n - 1$, he has a non-zero integer a_i , and he would like the following to hold:

- if $a_i > 0$, then the sum of the number of arrows in the first i positions pointing to the right and the number of arrows in the last $n - i$ positions pointing to the left should be at least a_i
- if $a_i < 0$, then the sum of the number of arrows in the first i positions pointing to the left and the number of arrows in the last $n - i$ positions pointing to the right should be at least $-a_i$

In other words, for every line between two consecutive elements, he either has a condition on the number of arrows pointing **towards** it, or on the number of arrows pointing **away from** it.

Help Nežmah find an array that satisfies his conditions!

Input

The first line contains a single integer t ($1 \leq t \leq 2 \cdot 10^4$), the number of test cases.

The first line of each test case contains a single integer n ($2 \leq n \leq 3 \cdot 10^5$).

The second line of each test case contains the array a ($-n \leq a_i \leq n$, $a_i \neq 0$). The array a will always be such that there is a solution.

It is guaranteed that the sum of n over all test cases does not exceed $3 \cdot 10^5$.

Output

For each test case, output a string of length n , where the i -th character is $<$ if the arrow there faces left, and it's $>$ if it faces right. If there are multiple solutions, you can output any of them.

Example

standard input	standard output
7	<<>><<<
7	>><><<
4 -2 -3 5 -3 3	>>><
5	<<>><
4 -2 4 -2	<<<<<<
4	>>
-2 3 3	<<
5	
-2 -4 -1 3	
6	
1 2 -1 -2 -3	
2	
2	
2	
-1	

This page is intentionally left blank.

Problem B. Big Balls

Input file: **standard input**
 Output file: **standard output**
 Time limit: 4 seconds
 Memory limit: 1024 megabytes

Mr. Nežmah likes playing with balls. He has placed n balls on a line so that ball i is the i -th leftmost ball. Every ball has its weight, which we denote with w , and the direction it will move to, either left or right.

Once he is happy with his setup, all of Mr. Nežmah's balls simultaneously start rolling in their chosen direction. All neighboring balls are equally distant from each other, and all balls move with equal constant speed.

When two balls of weights w_i and w_j collide, they merge into one big ball. Its weight is equal to the sum of the weights of the balls that collided, $w_i + w_j$, and its direction corresponds to the direction of the heavier of the two balls. If both balls have the same weight, the resulting ball will move to the right.

Mr. Nežmah noticed that, if enough time passes to let all collisions happen, the remaining balls can be split into two sets – balls that move left and balls that move right. For both of those sets, he wants to know the sum of the weights of the balls in it. Notice that the sum of the two numbers will correspond to the sum of the weights of the initial balls.

He believes this problem is too easy, so he wants you to solve it independently for q different subarrays of balls. Specifically, in the i -th query, he gives you numbers l_i and r_i , and he wants you to answer his problem if we only consider the balls denoted with $l_i, l_i + 1, \dots, r_i$.

Input

The first line contains an integer n ($1 \leq n \leq 10^6$), the number of balls.

The next n lines contain descriptions of the balls. In the i -th line, there is an integer w_i ($1 \leq w_i \leq 10^9$), followed by a letter c_i , representing the i -th ball's weight and direction, respectively. c_i will be **L** if the ball initially moves to the left, or **R** if it moves to the right.

The next line contains an integer q ($1 \leq q \leq 10^6$), the number of Mr. Nežmah's queries.

The next q lines contain descriptions of the queries. In the i -th line there are integers l_i and r_i ($1 \leq l_i \leq r_i \leq n$), the endpoints of the subarray.

Output

For each query, output two integers: the sum of weights of balls that will end up moving to the left, and the sum of weights of balls that will end up moving to the right.

Example

standard input	standard output
5	3 0
3 R	6 0
1 R	0 6
2 L	0 11
3 L	6 2
2 R	
5	
2 3	
2 4	
1 3	
1 5	
2 5	

This page is intentionally left blank.

Problem C. Crooked Cycles

Input file: standard input
 Output file: standard output
 Time limit: 3 seconds
 Memory limit: 1024 megabytes

Mr. Nežmah has a peculiar taste in triples. He likes m unordered triples of pairwise distinct numbers (a_i, b_i, c_i) .

A *cycle* of length k is a sequence of (not necessarily distinct) numbers v_1, \dots, v_k where every three consecutive numbers form a triple Nežmah likes, as well as (v_{k-1}, v_k, v_1) and (v_k, v_1, v_2) .

A cycle is *crooked* if its length is **not** divisible by 3. Check whether there exists a crooked cycle.

Input

The first line contains a single integer t ($1 \leq t \leq 2 \cdot 10^4$), the number of test cases.

The first line of each test case contains a single integer m ($1 \leq m \leq 2 \cdot 10^5$).

The following m lines contain integers a_i, b_i , and c_i ($1 \leq a_i, b_i, c_i \leq 2 \cdot 10^5$). The numbers a_i, b_i , and c_i are pairwise distinct for each i , and all unordered triples are distinct.

It is guaranteed that the sum of m over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, output **yes** if there exists a crooked cycle; otherwise, output **no**.

Example

standard input	standard output
3	yes
4	yes
1 2 3	no
2 3 4	
3 4 1	
4 1 2	
8	
3 2 5	
6 3 7	
7 3 2	
1 5 6	
5 2 6	
7 4 6	
3 2 6	
3 2 1	
7	
7 4 6	
2 3 7	
7 2 4	
5 4 6	
3 6 7	
5 3 6	
1 4 6	

Note

In the second test case (2352632736) forms a crooked cycle (although there are smaller ones as well).

This page is intentionally left blank.

Problem D. Dangerous Data

Input file: **standard input**
 Output file: **standard output**
 Time limit: **2 seconds**
 Memory limit: **1024 megabytes**

Mr. Nežmah has started working as a secret agent for SOA. Evil Mr. M has a secret set of $n + k$ distinct integers, and SOA has already figured out n of them. Nežmah is currently in Mr. M's secret lair and has access to all of the $n + k$ integers. However, he can only memorize k integers, and he doesn't know which n integers SOA already knows. Help Nežmah construct k integers (not necessarily from Mr. M's set) to remember so that he can figure out what integers SOA is missing once he comes back.

This is a run twice problem.

In the first run, you are given n and k , and $n + k$ different integers. You have to output k integers.

In the second run, you are given n and k and n integers which form a proper subset of the $n + k$ integers given in the first run. The order may be shuffled. You are also given the k integers outputted in the first run **in the same order**. You should output the missing k integers from the first input.

Input

In the first run:

The first line contains the word **first** and the numbers n and k ($1 \leq n \leq 10^5$, $1 \leq k \leq 10$).

The second line contains $n + k$ integers a_i , the elements of Mr. M's set ($1 \leq a_i \leq 10^6$).

In the second run:

The first line contains the word **second** and the numbers n and k .

The second line contains k integers, the output of your first run.

The third line contains n integers b_i , the elements of Mr. M's set that SOA has already figured out ($1 \leq b_i \leq 10^6$).

Output

In the first run, output a single line with k integers m_i ($0 \leq m_i < 2^{20}$).

In the second run, output a single line with the k required elements in any order.

Examples

standard input	standard output
first 3 2 5 2 3 4 1	12345 0
second 3 2 12345 0 3 1 5	4 2

This page is intentionally left blank.

Problem E. Exchange Error

Input file: **standard input**
 Output file: **standard output**
 Time limit: 4 seconds
 Memory limit: 1024 megabytes

Mr. Nežmah has started his new gig as a quantitative researcher. He is working with historical data of an unnamed stock on an interval of n consecutive days. This data is represented as an array a_i , which indicates that the price of the stock **moved** by a_i on the i -th day. Just as he was going to start training his model, he noticed there was an error in the data!

The whole array a_i was shifted by an unknown constant. One of the key features in his model is the largest price change which happened in those n days. More formally, for an array b of length n , he is interested in

$$\max_{1 \leq l \leq r \leq n} b_l + \dots + b_r$$

Now, because the data is erroneous, Nežmah wants to calculate this value for different shifts of the array a , or formally

$$S(x) := \max_{1 \leq l \leq r \leq n} (a_l + x) + \dots + (a_r + x)$$

Help Nežmah calculate $S(x)$ for q different values of x !

Input

The first line contains a single integer t ($1 \leq t \leq 2 \cdot 10^4$), the number of test cases.

The first line of each test case contains two integers n and q ($1 \leq n, q \leq 2 \cdot 10^5$).

The second line of each test case contains the array a ($-10^8 \leq a_i \leq 10^8$).

The third line of each test case contains the q queries ($-10^8 \leq x \leq 10^8$).

It's guaranteed that the sum of n and the sum of q over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, output q numbers: the answers to the queries.

Example

standard input	standard output
2	18 14 -11 11 39 7 1
5 7	1 3 6 14 18 23 35 42
1 2 6 9 -4	
0 -1 -20 -2 5 -4 -8	
7 8	
6 -14 1 5 -14 3 -8	
-5 -3 0 4 5 6 8 9	

This page is intentionally left blank.

Problem F. Flipping Frenzy

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 1024 megabytes

Mr. Nežmah found a grid with n rows and m columns filled up with 0s and 1s. Due to his OCD, he wants to turn all the ones in the grid into zeros. He has k ordered pairs of numbers (a_i, b_i) he likes. The only operation he is allowed to do is the following:

- Choose a cell (r, c) either inside the first row or the first column.
- Choose one pair of numbers (a, b) he likes.
- Flip all the numbers inside the cells which are contained in the rectangle of height a and width b whose top left corner is the cell (r, c) . Flipping means changing all zeroes into ones, and ones into zeroes. The rectangle must be fully contained in the grid.

Help Nežmah clear the grid!

Input

The first line contains a single integer t ($1 \leq t \leq 300$) — the number of test cases.

The first line of each test case contains 3 integers n , m , and k ($2 \leq n, m \leq 1000, 1 \leq k \leq 20$), the dimensions of the grid and the number of pairs (a_i, b_i) .

The next n lines contain m characters each, that are all 0 or 1, describing the grid.

The next k lines each contain two integers a_i and b_i ($1 \leq a_i < n, 1 \leq b_i < m$, all pairs (a_i, b_i) are distinct).

It is guaranteed that the sum of n and the sum of m over all test cases do not exceed 1000.

Output

For each test case, output the number of operations ($0 \leq op \leq k(n + m)$).

In the next op lines, output 3 integers r_i , x_i , and y_i ($1 \leq r_i \leq k, 1 \leq x_i \leq n - a_{r_i} + 1, 1 \leq y_i \leq m - b_{r_i} + 1$, at least one of x_i and y_i is equal to 1), meaning that you choose to flip all cells inside the rectangle with a_{r_i} rows and b_{r_i} columns with its upper left corner at (x_i, y_i) .

A solution is guaranteed to exist, and if there are multiple solutions you can output any of them.

Example

standard input	standard output
5	5
6 6 2	1 1 2
000000	2 1 2
111110	2 2 1
000000	2 3 1
100001	2 4 1
011111	6
111110	1 1 1
5 5	1 1 2
3 5	1 4 1
4 5 2	2 1 3
10111	2 1 5
00101	2 3 1
10000	0
01100	10
1 3	1 1 1
2 1	1 1 2
2 2 1	1 2 1
00	2 1 1
00	2 1 2
1 1	2 4 1
5 5 3	3 1 1
10100	3 1 2
10010	3 1 4
00000	3 2 1
01101	5
00100	3 1 4
3 4	2 1 3
2 3	4 1 2
4 2	2 4 1
5 5 4	3 3 1
01010	
01010	
10101	
00100	
00100	
2 2	
2 3	
3 2	
3 3	

Problem G. Grid Gradient

Input file: standard input
Output file: standard output
Time limit: 3 seconds
Memory limit: 1024 megabytes

Mr. Nežmah has received a grid with n rows and m columns as a birthday present. He wants to fill it up with numbers from 1 to 4, and as he is obsessed with gradients, he wants the absolute difference between numbers written in cells sharing a side to be exactly one.

As he is even more obsessed with counting stuff, help him find the number of such grids modulo 998 244 353!

Input

The first line contains integers n and m ($1 \leq n, m \leq 24$).

Output

In a single line, output the number of such grids modulo 998 244 353.

Examples

standard input	standard output
1 2	6
3 2	34
7 10	657297226

Note

In the first example, there are exactly six ways to fill up such a grid. They are the following: $(1, 2)$, $(2, 1)$, $(2, 3)$, $(3, 2)$, $(3, 4)$, $(4, 3)$.

This page is intentionally left blank.

Problem H. Halfway Hawser

Input file: standard input
 Output file: standard output
 Time limit: 1 second
 Memory limit: 1024 megabytes

Mr. Nežmah found a large string of length n ; in fact, it is so large it would be more appropriate to call it a *hawser*. However, he would like to remove exactly one letter from it. From the n strings which can be obtained in such a way, he is interested in the lexicographic median.

Formally, let t_i denote the string obtained by removing the i -th letter from s . Let p be any permutation of numbers from 1 to n such that t_{p_i} is lexicographically smaller than or equal to $t_{p_{i+1}}$. He wants to know $t_{p_{\lfloor \frac{n+1}{2} \rfloor}}$.

Input

The first line contains a single integer t ($1 \leq t \leq 2 \cdot 10^4$), the number of test cases.

The only line of each test case contains a single string s ($2 \leq |s| \leq 10^6$), consisting only of lowercase english characters.

The sum of $|s|$ across all test cases is at most 10^6 .

Output

For each test, in a single line, output the required string.

Example

	standard input	standard output
	4	zagrb
	zagreb	klokan
	klokani	oblutc
	oblutci	ac
	abc	

This page is intentionally left blank.

Problem I. Intricate Instrument

Input file: **standard input**
 Output file: **standard output**
 Time limit: **2 seconds**
 Memory limit: **1024 megabytes**

Dorijan gifted Mr. Nežmah a very intricate instrument. The instrument can play m different notes numbered from 0 to $m - 1$. Interestingly, it can play a tune of n notes s_0, \dots, s_{n-1} only when n is **prime** and the following is true

- the sequence is increasing i.e. $s_i < s_{i+1}$ for all $0 \leq i < n - 1$
- there exist a, b, c and d such that $s_{(a+b \cdot i) \bmod n} = (c + d \cdot i) \bmod m$ for all $0 \leq i < n$, where $0 \leq a < n$, $1 \leq b < n$ and $0 \leq c < m$, $1 \leq d < m$.

Notice that such a, b, c and d uniquely define the given sequence. Nežmah is currently composing a new tune, and he has already filled up some positions with notes. Help him finish the tune! It is enough to output any a, b, c and d which define a valid sequence that matches in the positions Nežmah has already set. It is guaranteed that a solution exists.

Input

The first line contains a single integer t ($1 \leq t \leq 2 \cdot 10^4$), the number of test cases.

The first line of each test case contains 2 integers n and m ($3 \leq n \leq 3 \cdot 10^5$, n is prime, $n \leq m \leq 10^9$).

The second line of each test case contains the resulting array ($-1 \leq u_i < m$, where $u_i = -1$ if Nežmah hasn't yet decided which note should be played in this position).

It is guaranteed that the sum of n over all test cases does not exceed $3 \cdot 10^5$.

Output

For each test case, output 1 line with 4 integers: the required values of a , b , c , and d . If there are multiple possible answers, output any of them.

Example

standard input	standard output
8	3 4 5 7
7 12	2 2 7 8
0 2 4 5 7 9 11	2 1 5 3
5 19	0 1 2 5
-1 4 -1 12 -1	1 4 1 13
3 7	0 1 0 1
1 4 -1	0 1 0 1
7 35	1 3 1 9
2 -1 12 -1 -1 27 -1	
5 14	
0 -1 11 -1 -1	
3 6	
-1 -1 2	
3 16	
-1 -1 -1	
5 14	
-1 1 -1 9 -1	

This page is intentionally left blank.

Problem J. Jolly Janes

Input file: **standard input**
 Output file: **standard output**
 Time limit: 1 second
 Memory limit: 1024 megabytes

While working for SOA, Mr. Nežmah somehow ended up on the Croatian version of The Bachelor! In the new season of The Bachelors, there are two bachelors, Mr. Nežmah and Mr. M, and n girls competing for their love. The girls are numbered from 1 to n .

As everyone knows, to advance to the next round, a girl has to receive a rose from either Mr. Nežmah or Mr. M. During the dates, both of them have already given out some roses, and currently, Mr. Nežmah has A roses left and Mr. M has B roses left. The girls already given roses by Mr. Nežmah are represented by a string s where the i -th character is '1' if the girl has received a rose during the dates and '0' otherwise. Roses given out by Mr. M are analogously represented by the string t .

The final ceremony is about to start, and the girls are already panicking. They know their Bachelors don't really care about the roses given out during the ceremony, so each of them will randomly distribute their roses among the girls who haven't yet received a rose from them. They will always distribute all of the roses they have left. Please help the girls calculate the expected number of them who make it through to the next round!

Input

The first line contains integers n , A , and B ($1 \leq n \leq 10^5, 0 \leq A, B \leq n$).

The second line contains s , a string of length n containing only the characters '0' and '1'.

The third line contains t , a string of length n containing only the characters '0' and '1'.

Output

In a single line, output the expected number of girls who make it through.

Your answer is considered correct if its absolute or relative error doesn't exceed 10^{-6} . Namely, if your answer is a , and the jury's answer is b , then your answer is accepted if $\frac{|a-b|}{\max(1,|b|)} \leq 10^{-6}$.

Examples

standard input	standard output
3 1 2 010 000	2.6666666667
7 3 2 0010000 0111010	6.5000000000

This page is intentionally left blank.

Problem K. Kempt Kale

Input file: **standard input**
 Output file: **standard output**
 Time limit: **2 seconds**
 Memory limit: **1024 megabytes**

Mr. Nežmah has started a kale farm. He wants to grow a total of n pieces of kale in the following $2n$ days. On the i -th day, Nežmah can:

- plant a piece of kale with tastiness a_i ,
- harvest a piece of kale.

Of course, every time Nežmah decides to harvest a piece of kale, there must exist at least one piece that was planted on a prior day and hasn't been harvested yet.

The total tastiness is equal to the sum of the tastiness of all kale Nežmah harvests. Help Nežmah devise a plan to maximize the total tastiness!

Input

The first line contains a single integer t ($1 \leq t \leq 2 \cdot 10^4$) — 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 second line of each test case contains the array a ($1 \leq a_i \leq 10^9$).

It's guaranteed that the sum of n over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, output n numbers, the days Nežmah should plant kale. If there are multiple solutions, you can output any of them.

Example

standard input	standard output
3	1
1	1 3 4
4 3	1 3
3	
1 2 4 9 6 3	
2	
5 100 100 200	

Note

In the 1st test case, the maximum tastiness is 4.

In the 2nd test case, the maximum tastiness is $1 + 4 + 9 = 14$.

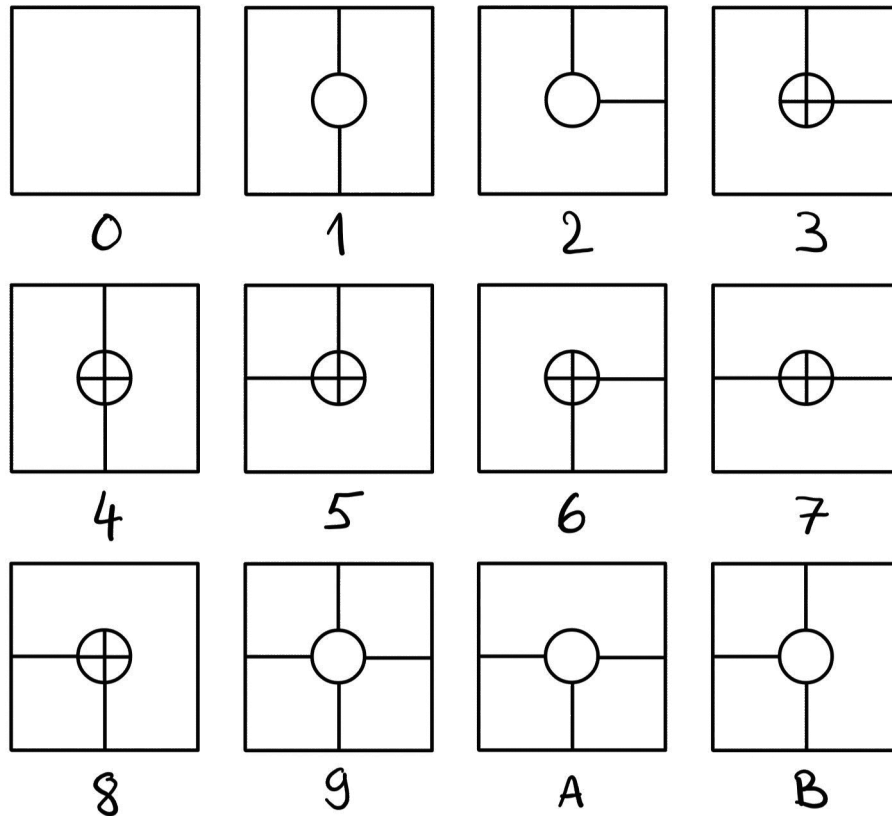
In the 3rd test case, the maximum tastiness is $5 + 100 = 105$. Note that 1, 2 would be accepted as well, as it gives the same sum.

This page is intentionally left blank.

Problem L. Loop Land

Input file: standard input
 Output file: standard output
 Time limit: 2 seconds
 Memory limit: 1024 megabytes

Mr. Nežmah received a puzzle for his birthday. The puzzle can be represented as a grid with n rows and m columns. Each cell can be one of 12 types, labeled with numbers from 0 to 9 and with A and B . Some of the cells are static, some can be rotated by specific degrees, and some function as switches. A detailed explanation of all the types follows this graphic:



Graphic of all types of cells

All cell types except 0 have a positive number of arms. The goal of the puzzle is to pair up all the arms inside the grid. The cells can be transformed according to their type.

- Cell types 0 and 9 are static and cannot be changed.
- Cell types 3, 4, 5, 6, 7, and 8 are switches, meaning that the **single** arm must be in exactly one of the two drawn positions.
- Cell types 1 and 2 can be rotated by 0° , 90° , 180° , and 270° .
- Cell types A and B can be rotated by 0° , 180° .

Nežmah wants to solve the puzzle, so he needs your help!

Input

The first line contains a single integer t ($1 \leq t \leq 2 \cdot 10^4$), the number of test cases.

The first line of each test case contains integers n and m ($1 \leq n, m \leq 10^3$).

The following n lines contain strings of length m describing the n rows of the grid. The j -th character in the i -th string denotes the type of the cell in the i -th row and j -th column.

It is guaranteed that the sum of $n \cdot m$ over all test cases does not exceed 10^6 .

Output

For each test case, output $2n - 1$ lines. Each line should contain a string of length $2m - 1$. If the cell in the i -th row and j -th column is of non-zero type, the character in the $(2i - 1)$ -th string on the $(2j - 1)$ -th position should be 'o'. Each 'o' should have a possible construction of arms going out of it (according to the type of the appropriate cell). Vertical arms are denoted with '|' and horizontal ones with '-'. All other characters should be '.'.

A solution is guaranteed to exist, and if there are multiple solutions, you can output any of them.

Example

standard input	standard output
3	o-o-o..
4 4
2A70	o-o-o.o
2A58
87A2	o.o-o-o
21A5
2 2	o-o-o-o
68	o.o
35	.
7 6	o.o
065258	..o-o.o-o.o
622143
B9BBB0	o.o-o.o.o.o
5B2B20
7B3A86	o-o-o.o-o..
432A12
282A15	o.o-o.o-o..

	o-o.o-o-o.o

	o.o.o-o-o-o

	o-o.o-o-o-o

Problem M. Maximum Mex

Input file: standard input
 Output file: standard output
 Time limit: 5 seconds
 Memory limit: 1024 megabytes

Mr. Nežmah has a tree with n nodes. On each edge (u_i, v_i) , there is a number a_i . Nežmah noticed an interesting property: every value of a_i appears on **at most six edges**.

For every pair of nodes u and v , let's define their *goodness* as the smallest non-negative integer that doesn't appear on any edges on the path from u to v . Find the maximum value of goodness across all pairs of nodes.

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 ($3 \leq n \leq 10^5$).

The following $n - 1$ lines of each test case contain u_i, v_i , and a_i ($1 \leq u_i, v_i \leq n, u_i \neq v_i, 0 \leq a_i < n$).

It's guaranteed that the sum of n over all test cases does not exceed 10^5 .

Output

For each test case, in a single line, output the maximum value of goodness across all pairs of nodes.

Example

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

This page is intentionally left blank.