KAIST 9th ACM-ICPC Mock Competition

Sponsored By:

DEVSISTERS

NAVER D2

Jane Street

STARTLINK

Rules

- This contest is KAIST 9th ACM-ICPC Mock Competition.
- This contest is sponsored by DEVSISTERS, NAVER D2, Jane Street, and STARTLINK.
- This contest starts at 14:00 and ends at 19:00, Oct 3rd 2019, Korean Standard Time (GMT +9).
- You can only participate in a team of three students.
- Any use of network is prohibited during the competition, except for submitting source codes and accessing language reference sites. Here are examples of allowed reference sites.
  - `C/C++ : http://en.cppreference.com/w/`
  - `Java  : https://docs.oracle.com/javase/8/docs/api/`
  - `Python: https://docs.python.org/`
  - `Kotlin: https://kotlinlang.org/docs/reference/`
- Each team can bring up to 25 pages of printed, A4-sized, single-sided reference materials for use during the competition.
- This contest consists of 11 problems.
- Problems are NOT sorted by difficulty.
- Solutions to problems submitted for judging are called *run*s. Each run is judged as accepted or rejected, and the team is notified of the results.
- Teams are ranked according to the most problems solved. Teams who solve the same number of problems are ranked first by least total time and, if need be, by the earliest time of submittal of the last accepted run.
- The *total time* is the sum of the time *consumed* for each problem solved. The time consumed for a solved problem is the time elapsed from the beginning of the contest to the submittal of the first accepted run plus 20 penalty minutes for every previously rejected run for that problem. There is no time consumed for a problem that is not solved.
- Memory limit for each problem is fixed as follows:
  - `C11/C++17       : 1024MB`
  - `Java/Kotlin     : 1536MB`
  - `PyPy 2/Python 3: 2048MB`
- Each problem may have different time limit.

Problem list

#	Problem Name	Time limit (All Languages)
A	6789	1000 ms
B	And the Winner Is... Ourselves!	4000 ms
C	Bigger Sokoban 40k	1000 ms
D	Capital	1000 ms
E	Gosu 2	2000 ms
F	Hilbert's Hotel	1500 ms
G	Lexicographically Minimum Walk	2000 ms
H	Maximizer	2000 ms
I	Minimum Diameter Spanning Tree	5000 ms
J	Parklife	1500 ms
K	Wind of Change	10000 ms

Problem A. 6789

Time limit: 1 second

Jaehyun likes digits. Among the 10 digits, 6, 7, 8, and 9 are his favorite. Therefore, he made a special card set consisting of only 6, 7, 8 and 9.

Currently, Jaehyun has $N \times M$ cards. Jaehyun wants to make a magical N by M matrix of cards. Each row of the matrix should contain M cards. He already arranged his cards in a shape of N by M matrix.

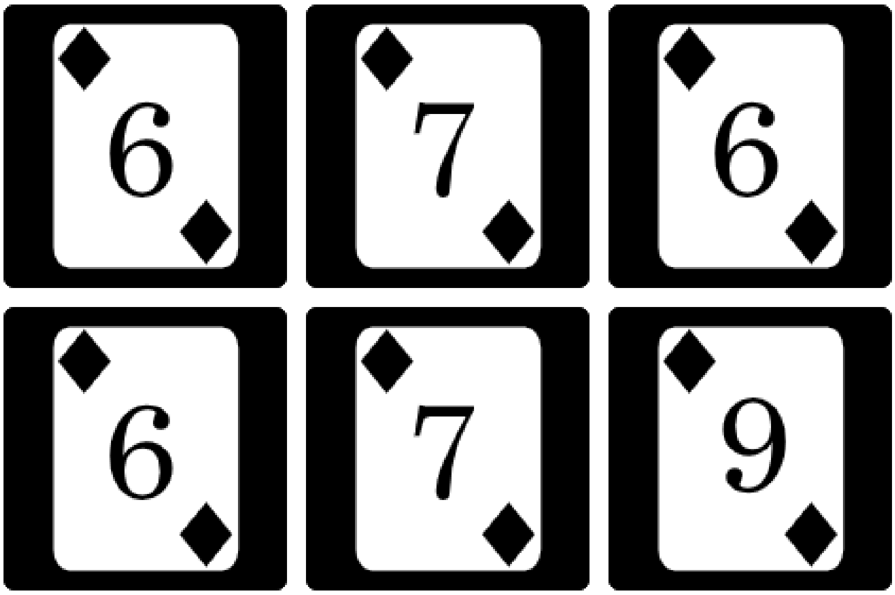


Figure 1. Initial state, not point symmetric.

To be a magic matrix, the matrix must be point symmetrical: Rotating the matrix 180 degrees results in the same original matrix. For example, 8 is point symmetrical with itself, and 6 and 9 are point symmetrical with each other. Jaehyun doesn't want to switch the position of the cards, so his goal is to make the matrix point symmetrical by only rotating the cards in their original positions.

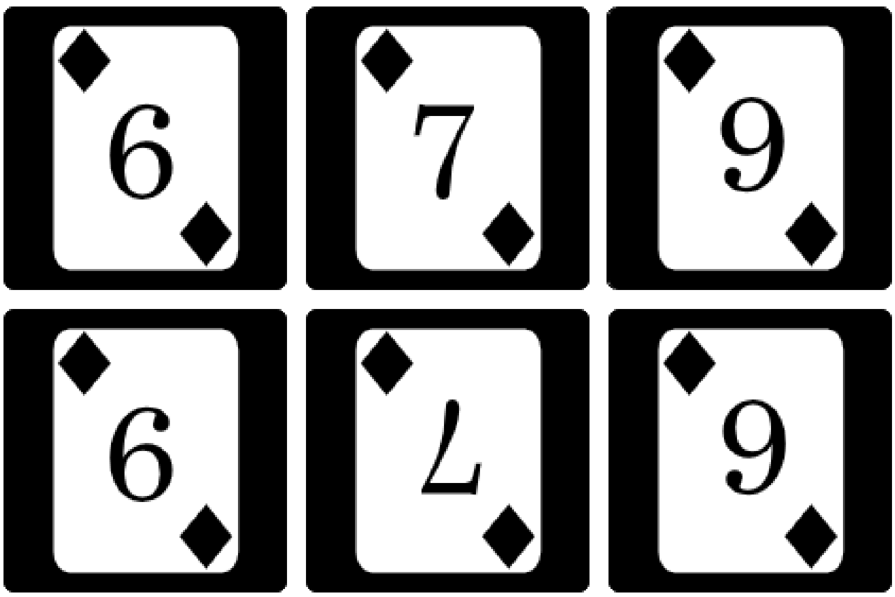


Figure 2. After rotating two cards, they are point symmetric.

Find the minimum number of cards you have to turn to make a magic matrix.

Input

The first line contains two integers, N and M. ($1 \leq N,\ M \leq 500$)

Each of the next N lines contains a string of M characters which denotes the numbers written in each card. It is guaranteed that each character is one of '6', '7', '8', or '9'.

Output

Print the minimum number of cards you have to turn to make a magic matrix in the first line. If it is not possible to make a magic matrix, print "-1". (without quotes)

Example

standard input	standard output
2 3 676 679	2
3 3 888 888 888	0
1 1 7	-1

Problem B. And the Winner Is... Ourselves!

Time limit: 4 seconds

Let us remind you about how the total penalties are calculated for this contest:

- When you solve a problem at T minutes, $T + 20V$ is added to your penalty, where V is the number of incorrect verdicts (except compile errors) received on that problem.
- If you do not solve a problem before the contest ends, the incorrect verdicts on that problem are not counted as penalties.

Here is a bad news for all of you: we, the problem setters, are planning to join the competition and solve our own problems!

We know our problems really well, so we can solve all the problems before the contest ends. Furthermore, we can precisely predict how long it takes to solve each problem, and how many incorrect verdicts (except compile errors) we get in each problem. Depending on the order of the problems we solve, our total penalty might differ. What is the minimum penalty if we solve all problems?

Input

11 lines are given as the input. The i-th line contains two space-separated integers, D_i and V_i, where D_i is the amount of minutes required to solve the i-th problem, and V_i is the number of incorrect verdicts on the i-th problem.

For each i, $1 \le D_i$ and $0 \le V_i \le 1\ 000$. Also, $\sum_{i=1}^{11} D_i \le 300$.

Output

Output the minimum penalty if we solve all problems.

Examples

standard input	standard output
20 1 20 0 20 3 10 0 10 0 10 0 30 0 30 0 30 0 20 0 20 10	1360

Note

The sample input does not necessarily reflect the actual difficulties of the problems.

The problem statement does not necessarily reflect the actual situation of the contest.

Problem C. Bigger Sokoban 40k

Time limit: 1 second

Sokoban is a famous puzzle game, where the player moves around in the $N \times M$-size grid, and pushes 1×1-size boxes to 1×1-size storage locations.

Bigger Sokoban is a possible variation of Sokoban, but the size of boxes and storage locations are bigger than 1×1. This problem especially uses 2×2 for both.

The rule of Bigger Sokoban is the same as Sokoban. Each square in the grid is an empty square or a wall. Some 2×2 area of empty squares contain 2×2-size box each and some 2×2 area of empty squares are marked as 2×2-size storage location each.

The player is in the grid and may move up, down, left, right to the adjacent empty squares, but should not go through walls, boxes, or outside of the grid. If the player tries to move into a box, it is pushed to the adjacent squares in that direction. Boxes must not be pushed to other boxes, walls, or outside of the grid, and they cannot be pulled. The number of boxes is equal to the number of storage locations. The puzzle is solved when all boxes are at the storage locations.

Your mission is to make a Bigger Sokoban grid that needs at least **40 000** moves to solve. To make the situation easier, the grid must satisfy the following constraints:

- $1 \leq N,\ M,\ N + M \leq 100.$
- The grid contains **one** box and **one** storage location.
- The player, the box, and the storage location must not intersect.

Input

There are no inputs for this problem.

Output

In the first line, print two space-separated integers $N,\ M$; they describe the size of the grid.

In each of the following N lines, print a string of length M; it describes each row of the grid. Each string must consist of `.`, `#`, `P`, `B`, `S`; each character means empty square, wall, player, box, storage location respectively.

The grid must contain exactly one `P`, exactly four `B`, and exactly four `S`. `B` and `S` each must form a 2×2 square. The grid, of course, must be solvable.

Note that the sample output is only to demonstrate a well-formatted output. Since it can be solved in less than 40 000 moves, it is not a correct answer.

Examples

standard input	standard output
<There are no inputs>	5 6 ....SS ....SS .#BB#. ..BB.P ......

Problem D. Capital

Time limit: 1 second

You are given N cities connected by M roads. Cities are numbered from 1 through N, and roads are numbered from 1 through M. For each pair of cities, there is a sequence of roads that connects those two cities. Road i has the length L_i kilometre and connects city A_i and city B_i bidirectionally. Every road has a positive length, so $L_i > 0$. Unfortunately, you have forgotten the length of each road.

You observed that, for each road, all people on road i are going from A_i to B_i, in a single direction. So, you assumed the hypothesis as follows:

- There is a capital city called S.
- People are moving from the capital city to other cities.
- People try to move in the shortest path. So the length of the shortest path from S to A_i is less than or equal to the length of the shortest path from S to B_i.

Can you find the capital city S which meets the criteria when you can assign the length of each road to be any positive real number? You may assume that there is at least one city that meets the criteria.

Input

The first line of the input contains two integers N ($2 \le N \le 500$) and M ($N - 1 \le M \le \frac{N(N-1)}{2}$).

In the i-th line of next M lines, A_i and B_i are given. ($1 \le A_i,\ B_i \le N$)

There are no loops or multiple edges. Formally, $A_i \ne B_i$, and $\{A_i,\ B_i\} = \{A_j,\ B_j\} \implies i = j$.

Output

In the first line, print the number of possible capital cities, K.

In the second line, print K space-separated integers which denotes all possible cities for the capital, in increasing order.

Example

standard input	standard output
2 1 1 2	1 1

Problem E. Gosu 2

Time limit: 2 seconds

Ho is an expert in martial arts called *Taebo*. She runs a Taebo school, and there are N students in her school. To increase the inner competition inside the Taebo school, she is going to make a *Taebo ranking website* which assigns all students to a certain rank. To find a suitable rank, Ho made all $N(N-1)/2$ pairs of students do a Taebo matchup with each other. In a Taebo matchup, exactly one person wins the match, and another person loses the match. The outcome of Taebo matchups may not be very simple: For example, there might be a case that student A beats B, B beats C, and C beats A. Such situation would make the ranking assignment pretty complicated as there is no definite winner from those three students.

To overcome the issue, Ho will find a **standard ranking chain** and assign other students with respect to such a chain. A **standard ranking chain** of length K, is a sequence of K different students $S_1, S_2, \cdots, S_k$ such that S_i beats S_j if and only if $i < j$. In other words, S_1 can beat all other students in the chain, S_2 can beat all other students in the chain except S_1, S_3 can beat all other students in the chain except S_1, S_2, and so on, and S_k can beat no other student in the chain. Ho's website will assign other students based on such a chain, which will make the assignment easier.

Ho is not only an expert in Taebo, but she is a math genius too. Ho knows, that for any Taebo matchup, she can find the standard ranking chain of length $1 + \lfloor \log_2(N) \rfloor$, where $\log_2(N)$ is a base 2 logarithm. In other words, for any $k \geq 1$ such that $2^{k-1} \leq N$, Ho can find a standard ranking chain of such a length.

While Ho is very good at computer programming too, she is a little bit lazy, therefore she delegates her work to you. You should find a standard ranking chain of length exactly $1 + \lfloor \log_2(N) \rfloor$.

Input

In the first line, the number of test cases T is given. For each test case, the following instances are given:

In the first line, the number of students N is given.

In the i-th line of the next N lines, a string of N characters, s_i, consisting of `W`, `L`, and `-` is given. Let's denote the j-th character of s_i as $s_{i,\ j}$. $s_{i,\ j}$ is given as follows:

- $s_{i,\ j} =$ `-`, if $i = j$.
- $s_{i,\ j} =$ `W`, if student i won student j.
- $s_{i,\ j} =$ `L`, if student j won student i.

- $1 \leq T \leq 250\,000$
- $1 \leq N \leq 512$
- The sum of N^2 for all test cases does not exceed $2\,500\,000$.
- $s_{i,\ i} =$ `-` $(1 \leq i \leq N)$
- If $i \neq j$, then $s_{i,j} =$ `W` or $s_{i,\ j} =$ `L`. $(1 \leq i \leq N)$
- If $s_{i,\ j} =$ `W`, then $s_{j,\ i} =$ `L`. $(1 \leq i,\ j \leq N)$
- If $s_{i,\ j} =$ `L`, then $s_{j,\ i} =$ `W`. $(1 \leq i,\ j \leq N)$

Output

For each test case, print exactly $1 + \lfloor \log_2(N) \rfloor$ integers in a single line, denoting the students in a standard ranking chain in the order of their skills. It can be proved that such a chain exists for every possible input.

Examples

standard input	standard output
5 1 - 2 -W L- 3 -LW W-L LW- 4 -WLW L-WL WL-W LWL- 5 -WLLW L-LLW WW-LL WWW-W LLWL-	1 1 2 3 2 3 1 4 4 3 1

Problem F. Hilbert's Hotel

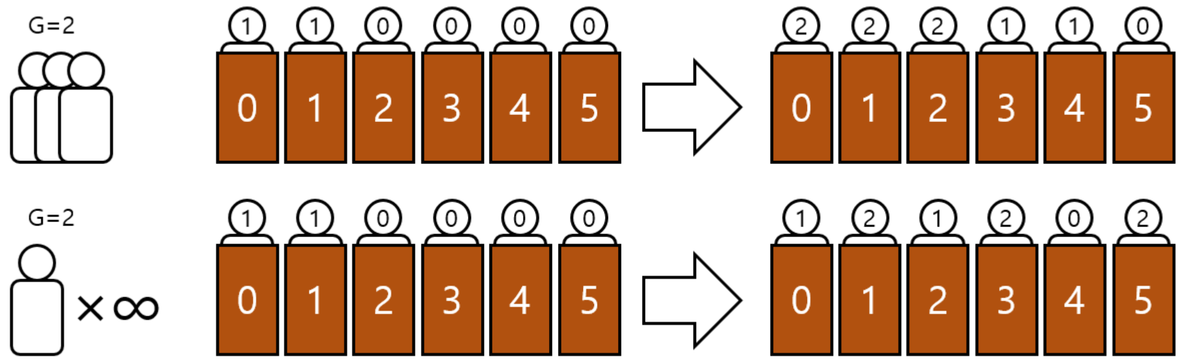
Time limit: 1.5 seconds

Hilbert's Hotel has infinitely many rooms, numbered 0, 1, 2, $\cdots$. At most one guest occupies each room. Since people tend to check-in in groups, the hotel has a group counter variable G.

Hilbert's Hotel had a grand opening today. Soon after, infinitely many people arrived at once, filling every room in the hotel. All guests got the group number 0, and G is set to 1.

Ironically, the hotel can accept more guests even though every room is filled:

- If k ($k \geq 1$) people arrive at the hotel, then for each room number x, the guest in room x moves to room $x + k$. After that, the new guests fill all the rooms from 0 to $k - 1$.
- If infinitely many people arrive at the hotel, then for each room number x, the guest in room x moves to room $2x$. After that, the new guests fill all the rooms with odd numbers.



You have to write a program to process the following queries:

- `1 k` - If $k \geq 1$, then k people arrive at the hotel. If $k = 0$, then infinitely many people arrive at the hotel. Assign the group number G to the new guests, and then increment G by 1.
- `2 g x` - Find the x-th smallest room number that contains a guest with the group number g. Output it modulo $10^9 + 7$, followed by a newline.
- `3 x` - Output the group number of the guest in room x, followed by a newline.

Input

In the first line, an integer Q ($1 \leq Q \leq 300\,000$) denoting the number of queries is given. Each of the next lines contains a query. All numbers in the queries are integers.

- For the `1 k` queries, $0 \leq k \leq 10^9$.
- For the `2 g x` queries, $g < G$, $1 \leq x \leq 10^9$, and at least x guests have the group number g.
- For the `3 x` queries, $0 \leq x \leq 10^9$.

Output

Process all queries and output as required. It is guaranteed that the output is not empty.

Examples

standard input	standard output
10 3 0 1 3 2 1 2 1 0 3 10 2 2 5 1 5 1 0 3 5 2 3 3	0 1 0 9 4 4

Note

If you know about "cardinals," please assume that "infinite" refers only to "countably infinite." If you don't know about it, then you don't have to worry.

Problem G. Lexicographically Minimum Walk

Time limit: 2 seconds

There is a directed graph G with N nodes and M edges. Each node is numbered 1 through N, and each edge is numbered 1 through M. For each i ($1 \le i \le M$), edge i goes from vertex u_i to vertex v_i and has a unique color c_i.

A **walk** is defined as a sequence of edges $e_1, e_2, \cdots, e_l$ where for each $1 \le k < l$, v_{e_k} (the tail of edge e_k) is the same as $u_{e_{k+1}}$ (the head of edge e_{k+1}). We can say a walk $e_1,\ e_2,\ \cdots,\ e_l$ starts at vertex u_{e_1} and ends at vertex v_{e_l}. Note that the same edge can appear multiple times in a walk.

The **color sequence** of a walk $e_1,\ e_2,\ \cdots,\ e_l$ is defined as $c_{e_1},\ c_{e_2},\ \cdots,\ c_{e_l}$.

Consider all color sequences of walks of length at most 10^{100} from vertex S to vertex T in G. Write a program that finds the lexicographically minimum sequence among them.

Input

The first line of the input contains four space-separated integers N, M, S, and T ($1 \le N \le 100\,000$, $0 \le M \le 300\,000$, $1 \le S \le N$, $1 \le T \le N$, $S \ne T$).

Then M lines follow: the i ($1 \le i \le M$)-th of them contains three space-separated integers u_i, v_i and c_i ($1 \le u_i, v_i \le N$, $u_i \ne v_i$, $1 \le c_i \le 10^9$); it describes a directional edge from vertex u_i to vertex v_i with color c_i.

The graph doesn't have multiple edges or loops, and each edge has a unique color. Formally, for any $1 \le i < j \le M$, $c_i \ne c_j$ and $(u_i,\ v_i) \ne (u_j,\ v_j)$ holds.

Output

If there is no walk from vertex S to vertex T, print "`IMPOSSIBLE`". (without quotes)

Otherwise, let's say $a_1,\ a_2,\ \cdots,\ a_l$ is the lexicographically minimum sequence among all color sequences of length at most 10^{100} from vertex S to vertex T.

- If $l \le 10^6$, print $a_1,\ a_2,\ \cdots,\ a_l$ in the first line. There should be a space between each printed integer.
- If $l > 10^6$, print "`TOO LONG`". (without quotes)

Example

standard input	standard output
3 3 1 3 1 2 1 2 3 7 1 3 5	1 7
3 4 1 3 1 2 1 2 1 2 2 3 7 1 3 5	TOO LONG
2 0 2 1	IMPOSSIBLE

Note

Sequence $p_1, p_2, \cdots, p_n$ is lexicographically smaller than another sequence $q_1, q_2, \cdots, q_m$ if and only if one of the following holds:

- There exists a unique j $(0 \leq j < \min(n, m))$ where $p_1 = q_1$, $p_2 = q_2$, $\cdots$, $p_j = q_j$ and $p_{j+1} < q_{j+1}$.
- $n < m$ and $p_1 = q_1$, $p_2 = q_2$, $\cdots$, $p_n = q_n$. In other words, p is a strict prefix of q.

Problem H. Maximizer

Time limit: 2 seconds

Maximizer has two permutations $A = [a_1, a_2, \cdots, a_N]$ and $B = [b_1, b_2, \cdots, b_N]$. Both A, B have length N and consists of **distinct integers** from 1 to N.

Maximizer wants to maximize the sum of differences of each element, $\sum_{i=1}^{N} |a_i - b_i|$. But he can only swap two adjacent elements in A. Precisely, he can only swap a_i and a_{i+1} for some i from 1 to $N-1$. He can swap as many times as he wants.

What is the minimum number of swaps required for maximizing the difference sum?

Input

The first line contains an integer N. ($1 \le N \le 250\,000$)

The second line contains N integers $a_1, a_2, \cdots, a_N$ ($1 \le a_i \le N$).

The third line contains N integers $b_1, b_2, \cdots, b_N$ ($1 \le b_i \le N$).

Each of $[a_1, a_2, \cdots, a_N]$ and $[b_1, b_2, \cdots, b_N]$ is a permutation. In other words, it is consisted of distinct integers from 1 to N.

Output

Print an integer, the minimum number of swaps required for maximizing the difference sum.

Example

standard input	standard output
3 1 2 3 1 2 3	2
4 3 4 1 2 3 2 4 1	3

Problem I. Minimum Diameter Spanning Tree

Time limit: 5 seconds

You are given a simple connected undirected weighted graph G with N nodes and M edges. Each node is numbered 1 through N.

A spanning tree of G is a subgraph of G, which is a tree and connects all the vertices of G. The diameter of a tree is the length of the longest path among the paths between any two nodes in the tree. A minimum diameter spanning tree of G is a spanning tree of G that has a minimum diameter.

Write a program that finds any minimum diameter spanning tree.

Input

The first line of the input contains two integers N ($2 \le N \le 500$) and M ($N-1 \le M \le \frac{N(N-1)}{2}$).

Then M lines follow: The i ($1 \le i \le M$)-th line contains three space-separated integers u_i, v_i and l_i ($1 \le u_i, v_i \le N$, $1 \le l_i \le 10^9$); it describes a bidirectional edge connecting vertex u_i and vertex v_i with length l_i.

It is guaranteed that the given graph doesn't have any loops or multiple edges, and the graph is connected.

Output

In the first line, print the diameter of the minimum diameter spanning tree of G.

In the next $N-1$ lines, print the description of the edges in the minimum diameter spanning tree of G. The j ($1 \le j \le N-1$)-th line should contain two space-separated integers x_i and y_i ($1 \le x_i,\ y_i \le N$); it describes a bidirectional edge connecting vertex x_i and y_i.

If there are several possible answers, print any one of them.

Examples

standard input	standard output
3 3 1 2 1 2 3 1 3 1 1	2 1 2 3 1
6 7 1 2 10 2 3 20 1 3 30 3 4 1000 4 5 30 5 6 20 4 6 10	1060 3 4 6 4 5 6 2 3 1 2

Problem J. Parklife

Time limit: 1.5 seconds

Figure 1. Gapcheon and an Expo bridge in a cloudy day

Gapcheon is a stream that flows through the *Daedeok Innopolis*: A research district in Daejeon which includes KAIST, Expo Science Park, National Science Museum, among many others. The waterfront of Gapcheon is used as a park, which is a facility for leisure and recreation.

In this problem, we model the Gapcheon as a slightly curved arc. In the arc, there are exactly 10^6 points marked by each centimeter. In Gapcheon, there are N bridges that connect two distinct points in the arc in a straight line segment. Such a line segment may touch other segments in an endpoint but never crosses them otherwise. For each pair of points, there exists at most one bridge that directly connects those two points.

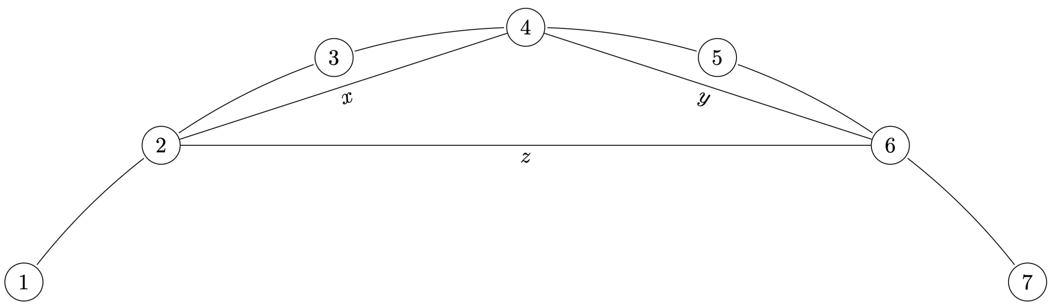


Figure 2. x, y, z are bridges that do not cross but only touch each other in an endpoint. This can be a possible input instance. Points with number $8 \ldots 10^6$ are omitted for brevity.

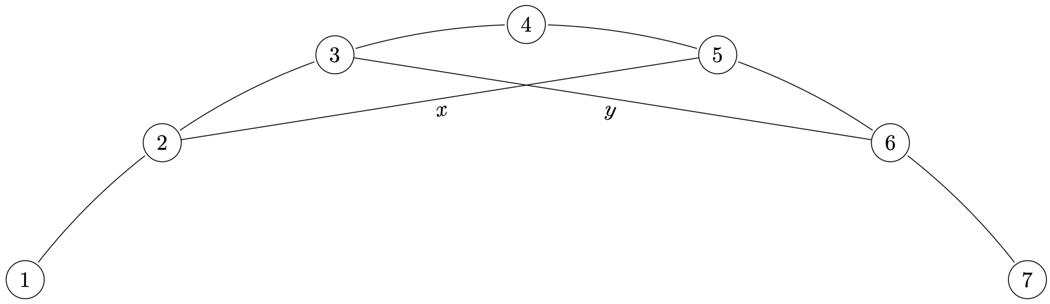


Figure 3. x, y are bridges that cross each other. This is not a possible input instance. Points with number $8 \ldots 10^6$ are omitted for brevity.

The city council is planning to place some lights in the bridges, to make Gapcheon as a more enjoyable place in the night. For each bridge, the city council calculated the aesthetical value if the lights are installed in these bridges. These value can be represented as a positive integer.

However, too many lightings will annoy the residents at midnight. To address this issue, the council decided to make some regulations: for every arc between two adjacent points, there should be at most k lighted bridges visible from there. We call a line segment **visible** from an arc connecting $i, i+1$, when one endpoint of the segment has an index at most i, and another endpoint of the segment has an index at least $i+1$.

The city council wants to consider the tradeoff between light pollution and the night view, so you should provide the maximum possible sum of aesthetical value, for all integers $1 \leq k \leq N$.

Input

The first line contains an integer N. ($1 \leq N \leq 250\,000$)

The next N lines contain three integers S_i, E_i, V_i, which denotes there is a straight line bridge connecting points S_i, E_i, and having aesthetic value V_i. ($1 \leq S_i < E_i \leq 10^6, 1 \leq V_i \leq 10^9$).

It's guaranteed that no lines connect the same pair of points, and no two different line segments cross.

Output

Print N integers separated by a space. The i-th integer ($1 \leq i \leq N$) should be the answer if $k = i$.

Examples

standard input	standard output
6 1 2 10 2 3 10 1 3 21 3 4 10 4 5 10 3 5 19	41 80 80 80 80 80
4 1 5 1 2 5 1 3 5 1 4 5 1	1 2 3 4

Note

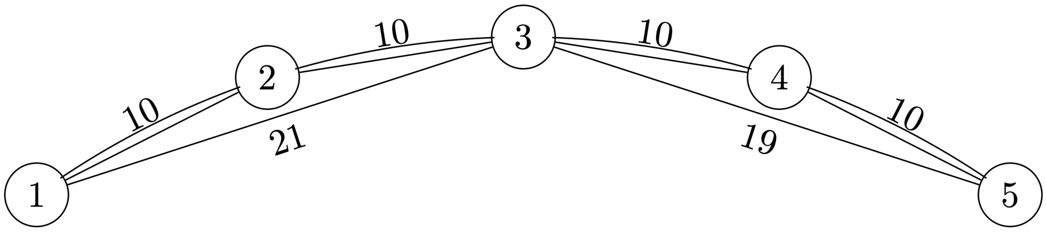


Figure 4. Depiction of Sample Input 1.

Copyright notice for *Figure 1*: *사진제공(한국관광공사 김지호)-한국관광공사*

Problem K. Wind of Change

Time limit: 10 seconds

The original title of this problem is "Tree Product Metric Voronoi Diagram Query Without One Point".

You are given two weighted trees $T_1,\ T_2$ of size N, where each vertex are labeled with numbers $1 \dots N$. Let $dist(T_1,\ i,\ j)$ be the total weight of the shortest path from node i to j in tree T_1, and define $dist(T_2,\ i,\ j)$ similarly.

Consider a point set of size N. Similar to Manhattan metric (in fact, this is a generalization of it), we can define the distance between two points $1 \leq i,\ j \leq N$: It is the sum of two distances, $dist(T_1,\ i,\ j) + dist(T_2,\ i,\ j)$. For each $1 \leq i \leq N$, please determine the "closest point" from the point i. Formally, for each i, you should find $min_{j \neq i} dist(T_1,\ i,\ j) + dist(T_2,\ i,\ j)$.

Input

In the first line, a single integer N denoting the number of vertices in both trees is given. $(2 \leq N \leq 250\,000)$

In the next $N-1$ lines, description of the first tree is given. Each of the $N-1$ lines contains three integers $S_i,\ E_i,\ W_i$, which indicates there is an edge connecting two vertices $S_i,\ E_i$ with weight W_i $(1 \leq S_i,\ E_i \leq N,\ 1 \leq W_i \leq 10^9)$

In the next $N-1$ lines, description of the second tree is given in the same format.

Output

Print N lines, each containing a single integer. In the i-th line, you should print a single integer denoting the answer for the point i.

Examples

standard input	standard output
5 1 2 10 2 4 20 3 4 30 4 5 50 1 2 15 1 3 25 1 4 35 1 5 25	25 25 85 65 105
9 5 7 6577 4 5 8869 5 9 9088 2 1 124 6 2 410 2 8 8154 4 8 4810 3 4 4268 3 9 763 6 2 8959 7 4 7984 3 8 504 8 6 9085 5 2 4861 1 9 8539 1 7 7834	18084 9369 9582 23430 26694 9369 23430 9582 22988