

Problem A. 6789

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

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.

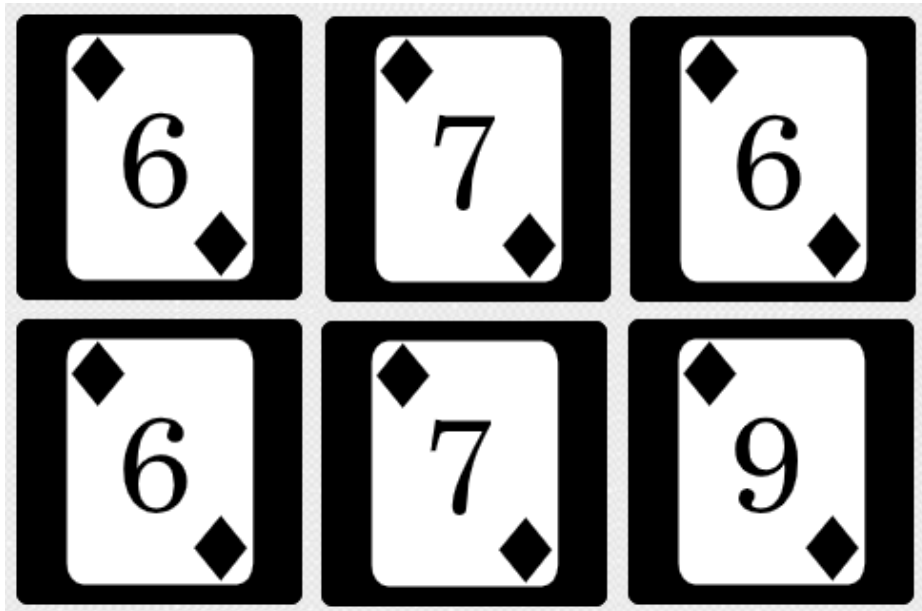


Figure 1. Initial state, not point symmetric.

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.

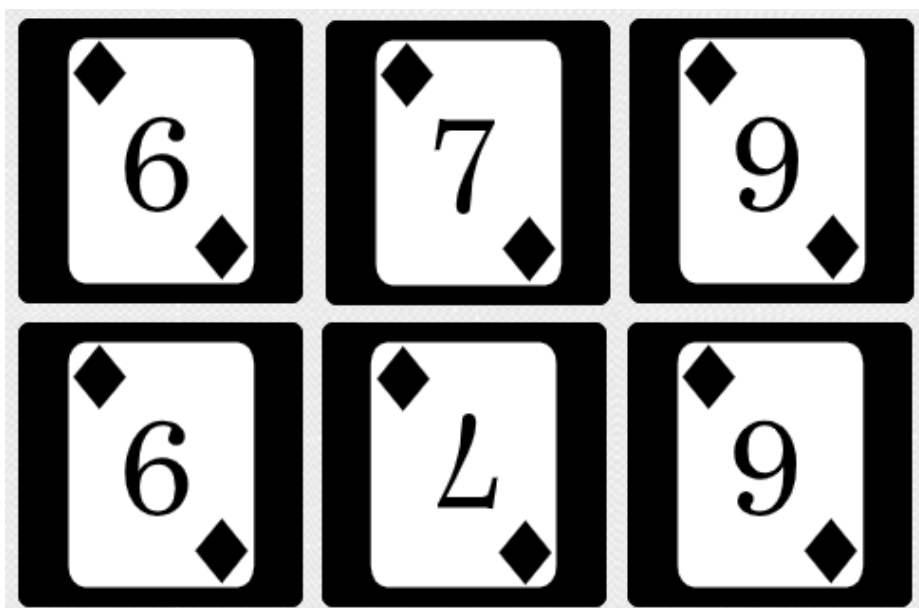


Figure 2. After rotating two cards, they are point symmetric.

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)

Examples

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

Problem B. Bigger Sokoban 40k

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

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 is no input 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 \cdot , $\#$, 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.

Example

standard input	standard output
	5 6SSSS .#BB#. ..BB.P

Problem C. Cleaning

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

Minje operates a game arcade. One of the games is situated in a grid of size $N \times M$. In each cell, one of the four characters L, R, U, D is written. This denotes the left/right/up/down direction, respectively. A player can freely move inside the grid, but he can **NOT** move toward the direction written in his cell, or outside of the grid.

Minje cleans the grid after the player enters and leaves the grid. However, cleaning all $N \times M$ cells is tiring. So Minje wants to only clean the cell which can be possibly visited by the player.

There are Q players who participated in the game. Minje remembers the first cell and the last cell that each player had stood during the game. These cells are not necessarily at the corner or edge. Please calculate the number of cells which Minje should clean. Every game is independent and does not affect other games' results.

Input

In the first line, three integers N, M, Q are given, denoting the size of the grid and the number of players, respectively. ($1 \leq N, M \leq 1\,000, 1 \leq Q \leq 300\,000$)

In the next N lines, strings of size M are given, denoting the character written in each grid cell. Every character is one of these four letters: L, R, U, D.

In the next Q lines, four integers x_1, y_1, x_2, y_2 are given. This denotes that the player started the game in cell (x_1, y_1) , and ended the game in cell (x_2, y_2) . ($1 \leq x_1, x_2 \leq N, 1 \leq y_1, y_2 \leq M$)

Output

In each of the Q lines, print the single integer denoting the answer for the query, in the order of the input. The answer to the query is:

- 0 if it's impossible to move from cell (x_1, y_1) to cell (x_2, y_2) .
- Otherwise, print the number of cells that Minje should clean.

Example

standard input	standard output
5 5 5	0
DDDDD	14
RDDDL	20
RRDLL	14
RUUUL	5
UUUUU	
1 1 5 5	
2 2 5 5	
3 3 5 5	
4 4 5 5	
5 5 5 5	

Problem D. Container

Input file: *standard input*
Output file: *standard output*
Time limit: 2.5 seconds
Memory limit: 1024 mebibytes

Cki86201 Container Creator (CCC) makes two kinds of containers. One kind of container has 1-ton capacity, and another one has 2-ton capacity. CCC currently has N containers in a warehouse, which is arranged in a line. To ship the containers, it's good to arrange the containers that are shipped first at the front. Therefore, CCC is going to rearrange those N containers.

To rearrange the containers we use machines. The machine can pick two or three consecutive containers, and reverse their orders. The cost of using the machine is the total capacity of the selected container plus the base cost C . Therefore, reversing two containers $[2, 1]$ takes cost $3 + C$, and reversing three containers $[1, 1, 2]$ takes cost $4 + C$. The number C is given as an input.

Although this task is trivial to **cki86201**, he has to play **Overwatch** now. Please find the way of rearranging the containers using the minimum cost.

Input

In the first line, two integers N, C are given. ($1 \leq N \leq 500, 0 \leq C \leq 1000$)

In the next line, a string of size N consisting of two characters '1', '2' is given. This indicates the capacity of the containers stored in the warehouse, in the order of **current** arrangement.

In the next line, a string of size N consisting of two characters '1', '2' is given. This indicates the capacity of the containers stored in the warehouse, in the order of **desired** arrangement.

It is guaranteed that both strings contain an equal number of '1', and equal number of '2'.

Output

In the first line, print the number K denoting the number of times you used the machine.

In the next K lines, print two integers i, j ($1 \leq i < j \leq N, j - i \leq 2$), which indicates that you have reversed the $i, i + 1, \dots, j$ -th containers from the current state, using the machine.

Your answer will be considered correct if the solution is valid (i.e simulating it with the outputted order produces the desired arrangement) and it uses the minimum possible cost.

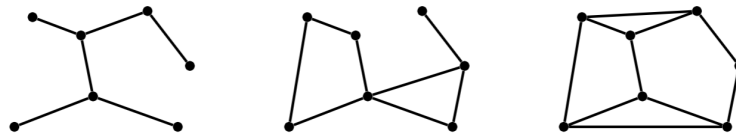
Examples

standard input	standard output
5 2 11221 21112	2 1 3 4 5
7 0 2212121 1212122	4 6 7 4 6 2 4 1 2
7 2 2212121 1212122	3 1 3 3 5 5 7

Problem E. Dead Cacti Society

Input file: *standard input*
Output file: *standard output*
Time limit: 10 seconds
Memory limit: 1024 mebibytes

Pet graphs are now a popular choice instead of animal pets in Korea. Among them, *Trees* are the most popular choice of pet graphs, due to their gentle properties and cute outlook. However, Koo is famous for being a maverick in this industry, because he only serves *Cactus* graphs to customers. *Tree* and *Cactus* are certain classes of graphs. A *tree* is a connected graph without cycles. A *cactus* is a connected graph, in which there exists no edge which belongs to two distinct cycles. Let's look at the example.

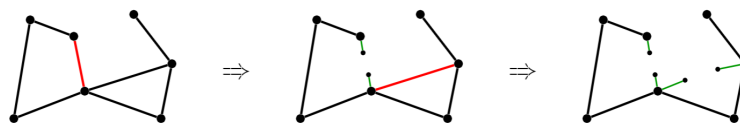


The graph in the first picture is both a tree and a cactus, the graph in the second picture is a cactus, and the graph in the final picture is neither of them. More cycles, more problems.

Koo had been serving Cactus graphs for 21 years with passion and love, but now he is old and tired. Although the popularity of pet graphs increased the visitors in his shop, they rarely became customers because most people considered Cactus graphs to be ugly for their pets. Koo's fortune is declining, and his life is getting tougher. In the end, Koo accepted the grim reality, and he decided to start a tree graph business - by cutting the edges of his lovely cacti.

A graph is a very sensitive creature, so cutting down the middle of the edge makes the whole graph decay. However, if you precisely cut the point where the vertex and the edge meet, the graph can heal itself. Thus, you should remove a whole edge, instead of cutting the middle. Also, you should never cut the graph to be disconnected: It's such a terrible thing to do to the graph.

If you cut the edge $e = \{u, v\}$, the vertices u, v will be wounded. A cactus is such a tenacious creature, so it can immediately heal the wounded scars, growing a new edge and a new end vertex. Every edge and vertex has a nonnegative **healing factor**, which is known to Koo in advance. If an edge e has a healing factor RE_e , and if an injured vertex v has a healing factor RV_v , then a new edge of length $RE_e + RV_v$ will be created in each of the wounded vertices, along with a new vertex in the opposite side of the edge.



In the picture, the red edge is the edge we will cut, and the green edges are the newly created edges.

If we cut exactly one edge from each cycle of a cactus, we can create a tree. Generally, people prefer trees which has a small diameter, where the diameter of the tree is a maximum possible length of some shortest path between the two vertices in the tree. To make a lot of money, Koo wants to cut the cactus and minimize its diameter. Given a cactus, help Koo to find the minimum possible diameter that can be obtained by cutting the edges in the cycle.

Input

In the first line, two integers N, M are given, denoting the number of vertices and edges of the cactus. ($3 \leq N \leq 100\,000$, $N \leq M \leq 150\,000$). Every vertex is labeled with number $1, 2, \dots, N$, and every edge is labeled with number $1, 2, \dots, M$.

In the next line, N integers RV_1, RV_2, \dots, RV_N are given, denoting the healing factor of each vertex. ($0 \leq RV_i \leq 10^9$).

In the next M lines, four integers A_i, B_i, L_i, RE_i are given in the i -th line. This indicates that the i -th edge is connecting two vertices A_i, B_i with edge length L_i , and has healing factor RE_i . ($1 \leq A_i, B_i \leq N, A_i \neq B_i, 1 \leq L_i, RE_i \leq 10^9$)

It is guaranteed that the given graph is a cactus, and there are no distinct edges connecting the same pair of vertices.

Output

Print a single integer denoting the answer.

Examples

standard input	standard output
3 3 1 2 3 3 1 2 3 1 2 1 2 2 3 3 1	10
5 6 1 2 3 4 5 1 2 6 1 1 3 5 4 2 3 4 2 1 4 3 6 1 5 2 3 4 5 1 5	22

Problem F. Hilbert's Hotel

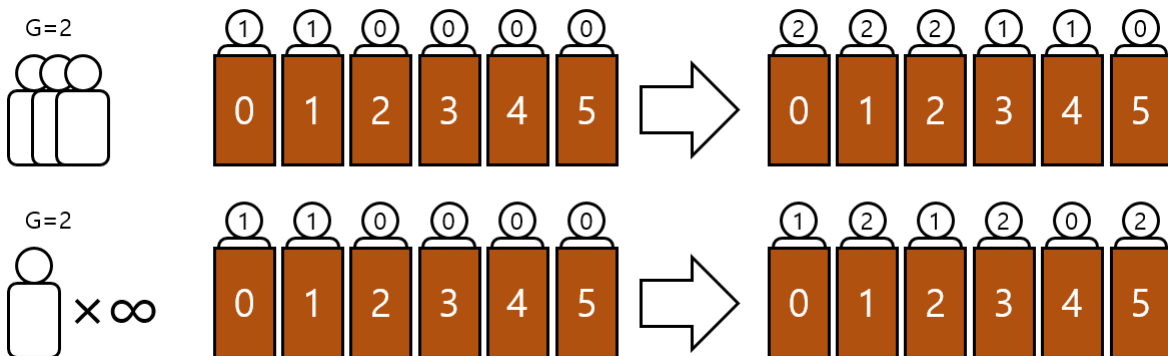
Input file: *standard input*
Output file: *standard output*
Time limit: 1.5 seconds
Memory limit: 1024 mebibytes

Hilbert's hotel has infinitely many rooms, numbered 0, 1, 2, ... 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.

Example

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

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

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

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 ($1 \leq i \leq M$), 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, \dots, e_l where for each $1 \leq k < l$, 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, \dots, 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, \dots, e_l is defined as $c_{e_1}, c_{e_2}, \dots, c_{e_l}$.

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 ($1 \leq N \leq 100\,000$, $0 \leq M \leq 300\,000$, $1 \leq S \leq N$, $1 \leq T \leq N$, $S \neq T$).

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

The graph doesn't have multiple edges and each edge has a unique color. Formally, for any $1 \leq i < j \leq M$, $c_i \neq c_j$ and $(u_i, v_i) \neq (u_j, v_j)$ 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, \dots, a_l is the lexicographically minimum sequence among all color sequences of length at most 10^{100} from vertex S to vertex T .

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

Examples

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, \dots, p_n is lexicographically smaller than another sequence q_1, q_2, \dots, q_m if and only if one

of the following holds:

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

Problem H. Maximizer

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

Maximizer has two permutations $A = [a_1, a_2, \dots, a_N]$ and $B = [b_1, b_2, \dots, b_N]$. 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 . ($1 \leq N \leq 250000$)

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

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

Each of $[a_1, a_2, \dots, a_N]$ and $[b_1, b_2, \dots, b_N]$ 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.

Examples

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

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

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 ($2 \leq N \leq 500$) and M ($N - 1 \leq M \leq \frac{N(N-1)}{2}$).

Then M lines follow: The i ($1 \leq i \leq M$)-th line contains three space-separated integers u_i , v_i and l_i ($1 \leq u_i, v_i \leq N$, $1 \leq l_i \leq 10^9$); 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 ($1 \leq j \leq N - 1$)-th line should contain two space-separated integers x_i and y_i ($1 \leq x_i, y_i \leq N$); 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

Input file: *standard input*
Output file: *standard output*
Time limit: 1.5 seconds
Memory limit: 1024 mebibytes



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.

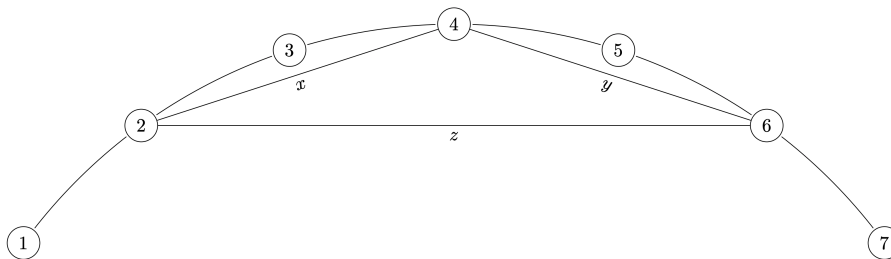


Figure 2. x, y, z are bridges that do not cross but only touch each other in an endpoint. This can be a possible input instance. Points with number $8 \dots 10^6$ are omitted for brevity.

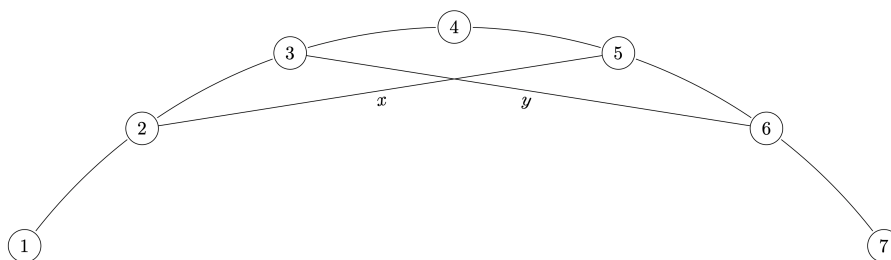


Figure 3. x, y are bridges that cross each other. This is not a possible input instance. Points with number $8 \dots 10^6$ are omitted for brevity.

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

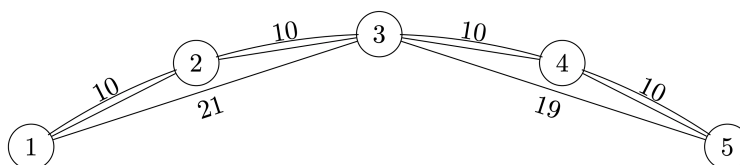


Figure 4. Depiction of Sample Input 1.

Problem K. Wind of Change

Input file: *standard input*
Output file: *standard output*
Time limit: 12 seconds
Memory limit: 1024 mebibytes

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 $\text{dist}(T_1, i, j)$ be the total weight of the shortest path from node i to j in tree T_1 , and define $\text{dist}(T_2, i, j)$ 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, $\text{dist}(T_1, i, j) + \text{dist}(T_2, i, j)$. For each $1 \leq i \leq N$, please determine the “closest point” from the point i . Formally, for each i , you should find $\min_{j \neq i} \text{dist}(T_1, i, j) + \text{dist}(T_2, i, j)$.

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 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