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.

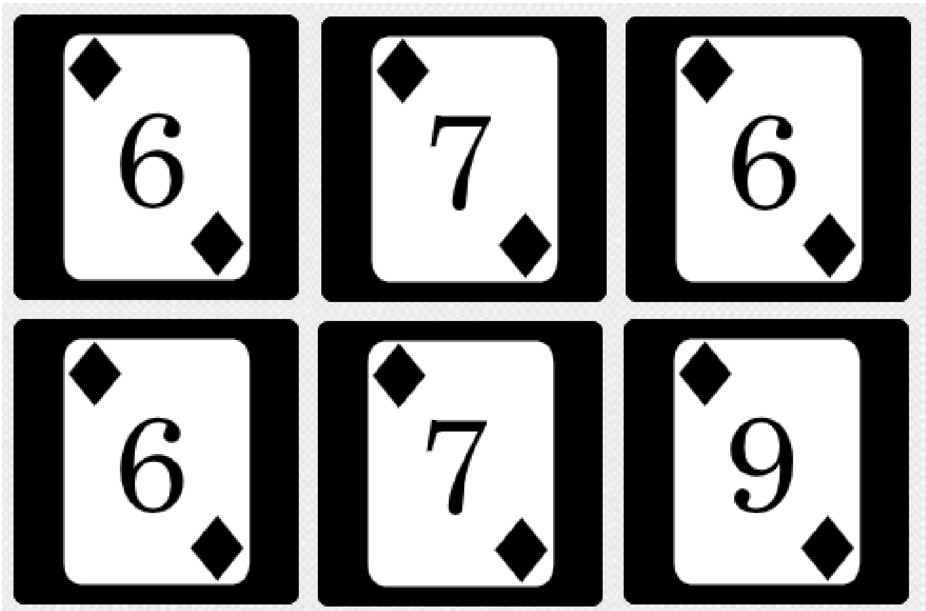


Figure 1. Initial state, not point symmetric.

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.

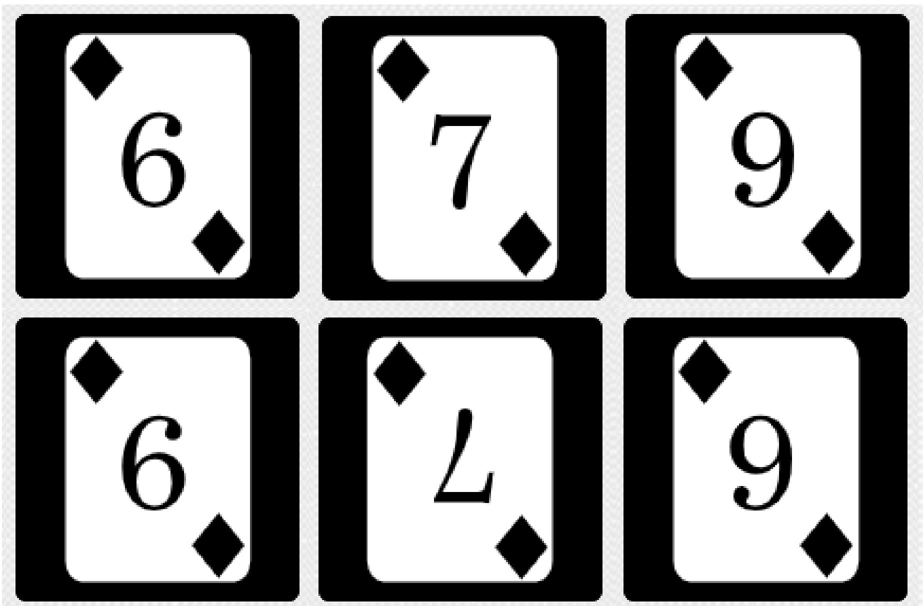


Figure 2. After rotating two cards, they are point symmetric.

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 F. Hilbert's Hotel

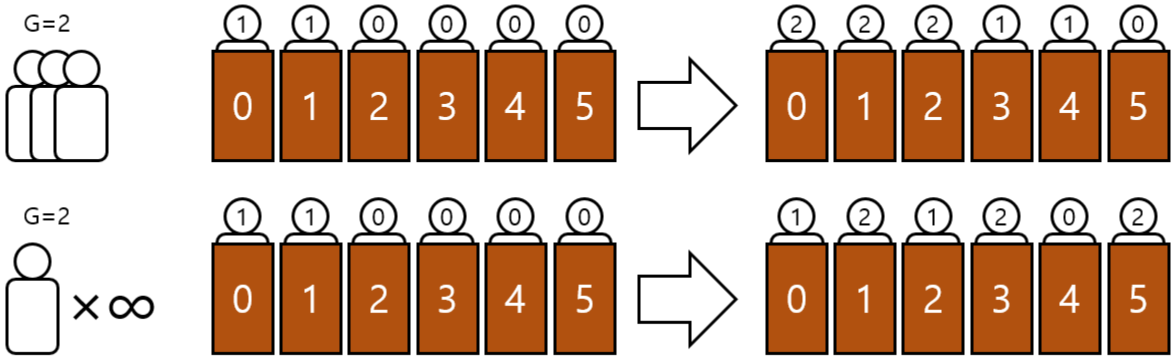
Input file:	*standard input*
Output file:	*standard output*
Time limit:	1.5 seconds
Memory limit:	1024 mebibytes

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 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

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 $(1 \le i \le M)$, 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 j $(1 \le j \le M)$-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 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)

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 $p_1, p_2, \cdots, p_n$ 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 $(0 \le j < \min(n, m))$ 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

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

Maximizer has two permutations $A = [a_1, a_2, \cdots, a_N]$ 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. $(1 \le N \le 250000)$

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.

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 ($2 \le N \le 500$) 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

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.

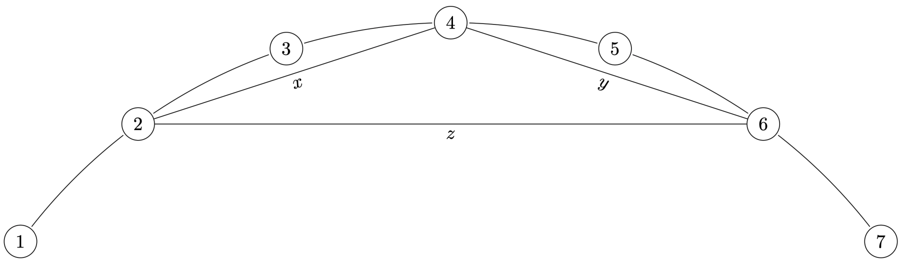


Figure 2. x, y, z are bridges that do not cross but only touch each other in an endpoint. This can be a possible input instance. Points with number $8 \dots 10^6$ are omitted for brevity.

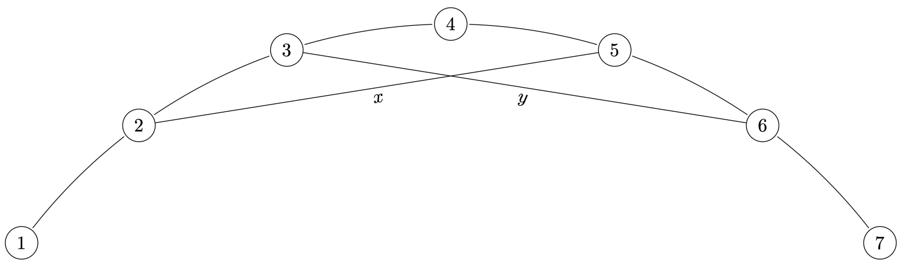


Figure 3. x, y are bridges that cross each other. This is not a possible input instance. Points with number $8 \ldots 10^6$ are omitted for brevity.

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 $1 \le k \le N$.

Input

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

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. ($1 \le S_i < E_i \le 10^6, 1 \le V_i \le 10^9$).

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 ($1 \le i \le N$) 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

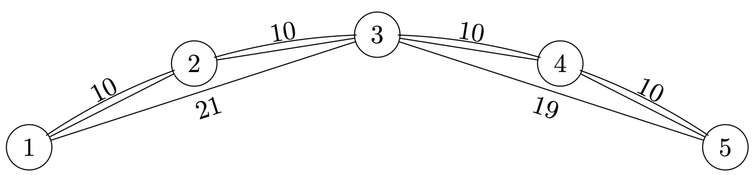


Figure 4. Depiction of Sample Input 1.

Problem L. Banquet

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

After the banquet tonight, you will leave the banquet venue together with the F new friends you have made and walk back to your hotel. You want to keep chatting with them as long as possible, though you may unfortunately not all be in the same hotel.

Everybody, you included, is taking one of the shortest paths back to their hotel. Bytesburg is made of N intersections numbered from 0 to $N-1$ and S streets linking two intersections. Streets can of course be walked in both directions. All hotels are located at intersections and are reachable from the banquet. The banquet takes place at intersection number 0.

Input

The input comprises several lines, each consisting of integers separated with single spaces:

- The first line consists of the three integers N, S, and F ($1 \le N \le 10^5$, $1 \le S \le 3 \cdot 10^5$, $0 \le F \le 10^4$).
- Each of the following S lines consists of three integers i, j, t, representing a street between intersections i and j, taking t minutes to walk through ($1 \le i, j \le N$, $1 \le t \le 10^6$).
- The last line consists of $F+1$ integers representing the intersections of the hotels where people stay; the first one is yours, the other ones are your friends respective hotels.

Output

The output should consist of a single line, whose content is an integer, the longest time (in minutes) you can stay accompanied by at least one friend.

Examples

standard input	standard output
7 12 1 0 1 1 0 2 2 0 2 4 1 3 2 2 3 1 1 4 2 4 2 2 1 5 3 2 5 1 5 6 2 4 6 2 3 6 3 3 6	2

Problem M. Buffet

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

Instead of a seated meal, some restaurants offer all-you-can-eat buffet lunches and dinners. This is usually a bargain for famished students. Bytica likes such buffet meals, but is always worried about how best to fill her plate with food. She values the n different items on the menu differently, and her goal is to have as much value as she can on her plate, constrained by the limited area of the plate and the limited availability of some menu items. Luckily, items on the menu are easily dividable and Alice can take an arbitrary fraction of each dish. Can you help her fill her plate?

Input

The input is formed of $n + 2$ lines:

- the first line consists of the number n of different items on the menu ($1 \le n \le 1000$);
- the second line consists of the area a of Alice's plate, an integer in mm^2 ($0 \le a \le 10^5$);
- each of the n remaining lines consists of information about a menu item, as two integers separated with a space; the first one is the value v_i of item i per mm^2, as perceived by Alice; the second one is the area a_i, in mm^2, that item i would occupy if Alice were to transfer it fully to her plate ($0 \le v_i \le 100$, $0 \le a_i \le 10^8$.

Output

A single line consisting of an integer: the maximal value that Alice can fit on her plate.

Examples

standard input	standard output
5 1000 50 230 80 12 10 1000000 25 450 2 50	26790

Problem N. Lucky Number

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

Consider the integer *lucky* if it either contains 7 in its decimal representation or is divisible by 7.

Given several integers, check which of them are lucky

Input

First line of the input contains one integer T — number of the test cases ($1 \le T \le 100$). Each of the next T lines contains one integer n ($-10^{100} \le n \le 10^{100}$).

Output

For each testcase print in the new line 1, if the integer is lucky, or 0 otherwise.

Example

```
4
7
105
17
108
```

Note 7 is lucky by two reasons (it contains 7 and it is divisible by 7); 105 is lucky becaus

Problem O. Counting Monuments

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

Alice loves visiting Bytesburg. Every time she is in the city, she tries to see as many monuments as she can, and every day of her visit she notes down in her travel notebook which monument she saw that day. After many years and many trips to Bytestburg, Alice is wondering how many different monuments she saw in her combined visits. She shows you her travel notebook, and asks for your help in computing this number.

Input

The input consists of n ($1 \leq n \leq 10^6$) lines, all of which contain, first, a date (in the format `YYYY-MM-DD`), then a space, then the name of a monument. Names of monuments are character strings of arbitrary length and containing arbitrary printable characters except for the newline character.

Output

The output should consist of a single line, whose content is an integer, the number m of distinct monuments in the input.

Examples

standard input	standard output
2016-12-30 Big Tower 2016-12-31 Old Byteland Castle 2016-12-31 Central Museum 2019-01-15 Big Tower 2019-01-15 Big tower 2019-01-15 Central museum 2019-01-16 Old Byteland Castle 2019-01-16 City Hall 2019-01-17 The Gigabyte Square	7

Problem P. Tourism

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

Every year, Seoul welcomes dozens of millions of tourists, who bring a considerable revenue to the city and its tourist industry. As an employee of the tourism bureau of the city of Seoul, you are tasked with computing how much money was spent by tourists on a particular week. You are given credit card receipts for that week and asked to tally them. As money is important, you want to obtain the exact number, to the cent.

Input

The input consists of n lines, all of which contain a real number representing the expense d in dollars ($1 \le n \le 10^6$, $0.00 \le d \le 10^6$). All real numbers are formatted as a non-empty sequence of decimal digits, a point (.) used as decimal separator, and two decimal digits for the fraction in cents.

Output

The output should consist of a single line, whose content is the **exact** value of total expenses, a decimal number in the same format as the input.

Examples

standard input	standard output
1532.23 226.09 1375.09 1227.20 2557.18	6917.79