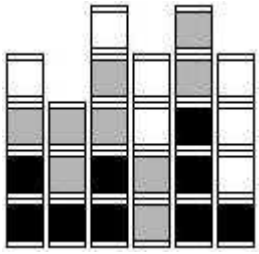
TASK	KONZERVE	PUTEVI
input	standard input	
output	standard output	
time limit (per test case)	3 seconds	1 second
memory limit (heap+stack)	64 MB	
points	**100**	**100**
	200	

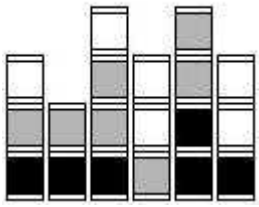
Mirko got a brand new air rifle for his birthday. To test its destructive power, he took cans of beans from his closet and rushed to the nearest playground.

He noted that there were three types of cans – black, white and grey, each containing different sorts of beans. Mirko lined up his cans in N columns, one behind another, so that the **black cans are on the bottom** of each column, **grey cans are in the middle**, with **white cans on top**. The columns may vary in height and not all columns need to contain cans of all three colours. However, the colours in a single column must be ordered as described.

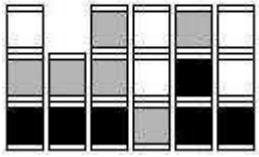
After arranging the cans, Mirko fires M bullets from his air rifle **horizontally**, each one at some height. The rifle is very powerful so the pellet, after shattering the first can at that height, continues flying in the same direction and destroys any cans at that height. After the bullet smashes a can, any cans that were on top of it fall down. The height of a shot is an integer; a shot at height 1 hits all cans lying on the ground, a shot at height 2 hits all cans immediately above those etc.



The layout in the first example



State after firing a shot at height 2



State after firing another shot, at height 4

In his game, Mirko awards himself 1 point for each black can shattered, 2 for each grey can, and 5 for each white can.

Write a program that, given the layout of the cans in columns and the heights of the shots in order, calculates the **number of points** received **for each shot**.

Input

The first line contains an integer N ($1 \leq N \leq 300000$), the number of columns.

The second line contains N non-negative integers, the number of black cans in each column.

The third line contains N non-negative integers, the number of grey cans in each column.

As one might imagine, the fourth line also contains N non-negative integers, the number of white cans in each column.

All numbers will be less than 10^6.

The fifth line contains an integer M ($1 \leq M \leq 300000$), the number of shots.

The sixth line contains M positive integers, the heights at which the shots are fired, in the order in which Mirko fires them. Each height will be less than 10^6.

Output

Output consists of exactly M lines, the number of points Mirko is awarded for each shot, in order in which he fires them.

Example test data

input

```
6
2 1 2 0 3 1
1 2 2 2 2 0
1 0 1 2 0 3
2
2 4
```

output

```
12
7
```

input

```
10
2 2 2 2 2 2 2 2 2 2
3 3 3 3 3 3 3 3 3 3
4 4 4 4 4 4 4 4 4 4
6
3 3 2 3 4 3
```

output

```
20
20
10
50
50
50
```

As you are bound to know by now, a **tree** is a connected graph consisting of N vertices and N−1 edges. Trees also have the property of there being exactly a single unique path between any pair of vertices.

You will be given a tree in which every edge is assigned a **weight** – a non-negative integer. The **weight of a path** is the **product** of the weights of all edges on the path. The **weight of the tree** is the **sum** of the weights of all paths in the tree. Paths going in opposite directions (A to B and B to A) are considered the same and, when calculating the weight of a tree, are counted only once.

Write a program that, given a tree, calculates its weight modulo 1 000 000 007.

Input

The first line contains the integer N ($2 \leq N \leq 100000$), the number of vertices in the tree. The vertices are numbered 1 to N.

Each of the following N−1 contains three integers A, B and W ($1 \leq A, B \leq N$, $0 \leq W \leq 1000$) describing one edge. The edge connects vertices A and B, and its weight is W.

Output

Output the weight of the tree, modulo 1 000 000 007.

Example test data

input

```
3
3 2 100
2 1 100
```

output

```
10200
```

input

```
4
1 2 5
1 3 5
1 4 5
```

output

```
90
```

input

```
5
1 2 2
2 3 3
4 3 2
5 3 2
```

output

```
55
```