CROATIAN OLYMPIAD IN INFORMATICS

April 14th, 2019

task	IZLET	LJEPOTICA	SEGWAY	TENIS
time limit	2 seconds	0.5 seconds	1 second	0.5 seconds
memory limit	512 MiB	512 MiB	512 MiB	512 MiB
points	100	100	100	100
	400			

When a competitive programmer thinks of a tree, it's not the same tree that a regural person thinks of. Luckily, in this task, both meanings are correct.

Nikola loves nature and often walks in the woods, looking at the trees and admiring their size, node degrees, paradoxically regular randomness, etc. In this day of lovely spring, there are even more reasons to look above into these magnificent creatures: the trees are full of color and that captured Nikola's attention.

So one day he observed a large tree with N nodes, seeing a color in each node. Was there a flower, an animal, or Nikola simply went mad, it's hard to say. But he was looking at that tree for a very long time, and in an N × N matrix he wrote, for each two nodes, the number of distinct colors on a unique simple path between (and including) these two nodes. Sadly, by admiring all those colors, Nikola completely forgot what the tree looked like and which colors were in the nodes.

So he needs your help. From the matrix he wrote, determine a possible tree and possible colors of the respective nodes. Colors should be denoted by numbers from {1, 2, …, N}. It is guaranteed that Nikola did not make a mistake; in other words, a solution will exist.

INPUT

The first line contains a subtask number (1, 2 or 3) from the *Scoring* section below.

The second line contains an integer N ($1 \leq N \leq 3000$), the number of nodes in the tree, which are denoted 1, 2, ..., N.

Each of the next N lines contains N integers, where *j*-th number in *i*-th line equals the number of distinct colors on the path from node *i* to node *j*.

OUTPUT

The first line should contain N space-separated integers between 1 and N: colors of the nodes 1, 2, ..., N.

Each of the next N – 1 lines should contain two integers A, B representing the edge (neighboring nodes) in the tree. Order of the edges and nodes inside an edge is arbitrary.

SCORING

subtask	points	additional constraints
1	18	all numbers in the matrix equal 1 or 2
2	25	there is a solution where the tree is a chain of nodes 1, 2, ..., N in that order, i.e., the edges are (i, $i + 1$) for each i = 1, ..., N – 1
3	57	no additional constraints

(See examples on the next page.)

SAMPLE TESTS

input

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

output

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

input

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

output

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

input

```
1
5
1 2 2 2 2
2 1 1 2 2
2 1 1 2 2
2 2 2 1 2
2 2 2 2 1
```

output

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

Beauty and the Geek is a reality television series advertised as connecting female beauties and male geeks with the goal of creating "The Ultimate Social Experiment". This task is advertised as connecting reality TV and competitive programming with the goal of creating a fun task.

Our hero is a beauty Ena, trapped in a complete binary tree of depth N. Each node of the tree has a value: root of the tree has a value of 1, and for each node with a value of x, its left child has a value of $2x$, and its right child has a value of $2x + 1$. Ena can move from a node to one of its two children, heading for the exit which is located in one of the leaves (nodes of depth N, with no children).

Ena knows an exact path from the root to the exit leaf. More precisely, she knows the correct sequence of N – 1 moves, each of them being "left" or "right", which would guide her from the root to the exit leaf. Unfortunately, Ena is not sure which side is left and which side is right. Therefore, during her trip, she changed her mind exactly K times about the meaning of "left" and "right". When she changes her mind, she moves accordingly until the end of the trip (a leaf node) or until the next change of mind. Ena's change of mind can happen only once before each move in the tree (including the first one). Also, nobody knows whether Ena had correct sides in mind while entering the root of the tree.

The producers of the TV show will save the lost Ena if you, her geek partner, answer correctly to the following question: ***What is the sum of leaf values where Ena can finish her trip, considering only leaves with values of at least A and at most B?***

INPUT

The first line contains integers N and K from the task description ($2 \le N \le 1000$, $0 \le K \le N - 1$).

In the second line there is a word containing N – 1 characters 'L' (left) and 'R' (right) representing the correct path from the root to the exit leaf.

The third line contains the number A from the task description, in binary form without leading zeros.

The fourth line contains the number B from the task description, in binary form without leading zeros.

Ena will be able to finish in leaves A and B.

OUTPUT

Output the required sum as a decimal integer modulo 1 000 000 007.

SCORING

subtask	points	additional constraints
1	8	K = 0
2	14	N ≤ 25
3	17	A is the smallest, and B is the greatest value where Ena can finish
4	61	no additional constraints

(See examples on the next page.)

SAMPLE TESTS

input	input	input
3 0 LR 101 110	4 2 LRR 1010 1110	5 2 RLLR 10010 10111
output	**output**	**output**
11	37	82

First sample description: Ena will never change her mind, but we don't known if she had the correct left/right sides in mind from the beginning. So, she might have followed the instructions correctly and go to the left, and then the right child. Or, she might have followed the inverse instructions, going first to the right child and then to the left child. The arriving leaves have values 5 and 6 which correspond to A and B, so the answer is 5 + 6.

Second sample description: Possible Ena's paths: (left, left, left), (left, left, right), (left, right, left), (right, left, right), (right, right, left), or (right, right, right).

There is a Segway race in the city of Dubrovnik. The race track consists of three sections, each of which is 100 meters long – therefore, the total length of the track is 300 meters. Based on the limitations of her/his Segway battery, each rider has a strategy: the speed at which he rides on the first 100 meters, the speed on the next 100 meters, and the speed on the last 100 meters, except when allowed to speed up the Segway to the maximum speed (explained in the next paragraph). Unfortunately, the Segways are very slow, taking between 1 and 50 seconds for each meter. Therefore, the speeds in this task are given in seconds per meter (instead of meters per second).

Along the track there are several acceleration points (accelerators). When a rider reaches an accelerator, his Segway gets extra power to ride at the maximum speed of 1 second per meter for the next **X mod 20** meters, where X is the number of riders strictly ahead of him at the moment he reached the accelerator (including those who have already completed the race). The rider is unable to use another accelerator before he has spent all extra power from the previous accelerator. At that moment, if there are no new accelerators, the rider continues to move at his default speed for the corresponding track section.

Assume that a rider will always use an available accelerator, even if it might not be the optimal strategy. An accelerator can be use by multiple riders, even at the same time. Your task is to write a program that simulates this race. Assuming that all Segway riders start at the same time, print the finish time for each rider in seconds.

INPUT

The first line contains an integer N ($2 \le N \le 20\,000$), the number of riders.

The K^{th} of the following N lines contains three integers between 1 and 50: the default speed of the K^{th} rider on the first 100 meters, the next 100 meters, and the last 100 meters of the track.

The next line contains an integer M ($0 \le M \le 299$), the number of acceleration points.

If M > 0, the following line contains a strictly increasing sequence of M integers between 1 and 299: the distances of the accelerators from the beginning of the track in meters.

OUTPUT

You should print N lines, where the K^{th} line contains the required time for the K^{th} rider.

SCORING

subtask	points	additional constraints
1	15	M = 1
2	40	$N \le 300$
3	45	no additional constraints

(See examples on the next page.)

SAMPLE TESTS

input

```
2
1 2 3
4 5 6
0
```

output

```
600
1500
```

input

```
3
5 5 5
6 2 10
10 9 2
2
100 199
```

output

```
1496
1799
2075
```

input

```
5
2 2 2
6 6 6
8 8 8
9 9 9
10 10 10
2
297 298
```

output

```
600
1790
2386
2676
2973
```

First sample description: There are no accelerators and both riders use their default speeds.

Second sample description: Rider #1 does not use the first accelerator (there is nobody ahead of him), but uses the second accelerator because rider #2 overtakes him in the meantime. Overall, rider #1 rides 299 meters for 5 seconds each, and 1 metar for 1 second.

Rider #2 uses the first accelerator (there is one rider ahead), but does not use the second one. Overall, he rides 100 meters for 6 seconds each, 1 meter for 1 second, 99 meters for 2 seconds each, and 100 meters for 10 second each.

Rider #3 after each accelerator rides 2 meters at maximum speed. Overall, he rides 100 meters for 10 seconds each, 2 meters for 1 second each, 97 meters for 9 seconds each, 2 meters for 1 second each, and 99 meters for 2 seconds each.

Third sample description: Of the two accelerators near the end of the track, rider #1 does not use any. Rider #2 uses both (for 1 meter) and then rides for another 1 meter at her default speed. Rider #3 uses the first accelerator (for 2 meters) and then rides for another 1 meter at her default speed. Riders #4 and #5 use extra power from the first accelerator all the way to the end of the track.

Vito loves tennis. Soon, he will organize a huge tournament with N participating players, denoted 1, 2, ..., N. Vito has followed the players' statistics in the last couple of years. Thus, he has determined their strenghts on three different court surfaces: clay, grass, and hardcourt. Namely, for each surface he has determined the players' ranking list, from the strongest to the weakest player on that surface.

The rules of Vito's tournament are quite unusual. A total of N – 1 matches will be played. In each match, two players that have not yet been eliminated will play against each other on a particular court surface. The player who is weaker on that surface will lose the match and thus be eliminated from the tournament. The only player who remains in the tournament after all N – 1 matches will be the winner of the tournament.

Vito is an influential man and can manipulate the outcome of the tournament. Namely, for each match of the tournament, Vito can choose both players and the court surface of the match. Of course, he can only choose players which have not been eliminated yet.

Vito often updates the statistics in his books in such a way that he sometimes swaps the positions of two players in a particular surface's ranking list. Besides, Vito has a lot of friends, some of which come to him with questions like this: *Player X is my nephew, is there any chance he will win the tournament* (*wink, wink*)? To answer their queries, Vito has made you an offer which is hard to refuse. You should write a program which will update the ranking lists and answer the questions of Vito's friends based on the ranking lists at that moment.

INPUT

The first line contains integers N and Q ($1 \leq N, Q \leq 100\ 000$), the number of players and the number of events.

Each of the next three lines contains a permutation of integers {1, 2, …, N} — the ranking of players on a particular court surface, from the strongest to the weakest one.

Each of the following Q lines is of one of the following types:

- "1 X", where $1 \leq X \leq N$ — Vito's friend wants to know if player X can win the tournament.
- "2 P A B", where $1 \leq P \leq 3$ and $1 \leq A \neq B \leq N$ — Vito has realised that he should swap the positions of players A and B in the P^{th} ranking list.

OUTPUT

For each query of type 1, output "DA" or "NE" (Croatian words for "YES" and "NO") in a separate line.

SCORING

subtask	points	additional constraints
1	7	$N \leq 15, Q \leq 10$
2	11	$N \leq 1000, Q \leq 10$
3	12	$Q \leq 10$
4	21	all events are of type 1
5	49	no additional constraints

SAMPLE TESTS

input

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

output

```
DA
DA
NE
```

input

```
6 7
4 6 1 5 3 2
5 1 4 2 6 3
4 6 1 5 2 3
1 2
2 2 4 5
1 1
2 2 4 5
2 2 5 6
1 2
1 1
```

output

```
DA
NE
NE
DA
```

First sample description:

If all matches are played on the first court surface, player 1 will win the tournament.

Example of a tournament where player 4 wins:

- Players 3 and 4 play on the third court surface – player 4 wins.
- Players 1 and 2 play on the first court surface – player 1 wins.
- Players 1 and 4 play on the third court surface – player 4 wins.

After updating the third court surface's ranking list (new ranking: 2 1 3 4), player 4 is the weakest on all surfaces and thus cannot win the tournament.