

Problem A. LCA begins

Input filename: `lca.in`
Output filename: `lca.out`
Time limit: 2 seconds
Memory limit: 256 Mb

Initially, there is a rooted tree consisting of the root (vertex with index 1) only. You need to answer queries of two following types:

- **ADD** $a\ b$ — add new vertex b with parent a (it is guaranteed that vertex a already exists);
- **GET** $a\ b$ — return lowest common ancestor of vertices a and b .

All vertex numbers will be from 1 to 1000.

Input file format

The first line of the input file contains a single integer k , the number of queries. The following k lines are the queries themselves. The number of queries of both types does not exceed 1000.

Output file format

For every **GET** query print one integer on its line.

Sample tests

<code>lca.in</code>	<code>lca.out</code>
9	1
ADD 1 2	1
ADD 1 3	1
ADD 2 4	2
GET 1 3	5
GET 2 3	
GET 3 4	
ADD 2 5	
GET 4 5	
GET 5 5	

Problem B. LCA continues

Input filename: `lca.in`
Output filename: `lca.out`
Time limit: 2 seconds
Memory limit: 256 Mb

Initially, there is a rooted tree consisting of the root (vertex with index 1) only. You need to answer queries of two following types:

- **ADD** $a\ b$ — add new vertex b with parent a (it is guaranteed that vertex a already exists);
- **GET** $a\ b$ — return lowest common ancestor of vertices a and b .

All vertex numbers will be from 1 to 500 000.

Input file format

The first line of the input file contains a single integer k , the number of queries. The following k lines are the queries themselves. The number of queries of both types does not exceed 500 000.

Output file format

For every **GET** query print one integer on its line.

Sample tests

<code>lca.in</code>	<code>lca.out</code>
9	1
ADD 1 2	1
ADD 1 3	1
ADD 2 4	2
GET 1 3	5
GET 2 3	
GET 3 4	
ADD 2 5	
GET 4 5	
GET 5 5	

Problem C. LCA minimum query

Input filename: `lca_rmq.in`
Output filename: `lca_rmq.out`
Time limit: 4 seconds
Memory limit: 256 Mb

You are given a rooted tree with $1 \leq n \leq 100\,000$ vertices, numbered from 0 to $n - 1$. You need to answer $1 \leq m \leq 10\,000\,000$ queries about the lowest common ancestor of pairs of vertices.

The queries are generated in the following way. You are given numbers a_1, a_2 and numbers x, y , and z . Numbers a_3, \dots, a_{2m} are generated as follows: $a_i = (x \cdot a_{i-2} + y \cdot a_{i-1} + z) \bmod n$. First query is $\langle a_1, a_2 \rangle$. If the answer to query $(i - 1)$ -th query is equal to v , then i -th query is $\langle (a_{2i-1} + v) \bmod n, a_{2i} \rangle$.

Input file format

The first line contains two numbers n and m . The root of the tree has index 0. The second line contains $n - 1$ integers, i -th of which is the parent of the i -th vertex. The third line contains two integers from 0 to $n - 1$: a_1 and a_2 . The fourth line contains three integers x, y , and z .

Output file format

Print a single integer — the sum of answers to all queries.

Sample tests

<code>lca_rmq.in</code>	<code>lca_rmq.out</code>
3 2 0 1 2 1 1 1 0	2

Problem D. Color Wizards

Input filename: `magic.in`
Output filename: `magic.out`
Time limit: 2 seconds
Memory limit: 256 Mb

The magical country consists of several cities, connected by two-way roads. Every town is reachable from every other. No road connects a city to itself, and no two cities are connected by two roads.

Two wizards live in the country: a blue one and a yellow one. The yellow wizard colours a road yellow after walking on it, and the blue one colours roads blue. However, if these two colours overlap, then the green colour is formed, which happens to be the least favourite colour of both wizards.

This year, a wizard conference is about to begin in the capital city f . Both wizards want to attend it, but also to turn the minimum number of roads green.

However, the initial location of the wizards is unknown in advance. They, therefore, ask you to solve this problem for k different initial positions.

Input file format

The first line contains two integers, $1 \leq n \leq 100\,000$ and $1 \leq m \leq 500\,000$ — the number of cities, and the number of roads in the magical country. The second line contains the index of the capital city $1 \leq f \leq n$. The following m lines contain road descriptions in the form $1 \leq a_i, b_i \leq n$. The next line contains a number $1 \leq k \leq 100\,000$ — the number of initial configurations to process. The following k lines contain two integers denoting the starting positions of both wizards.

Output file format

For every initial position print the minimum number of roads that will become green.

Sample tests

magic.in	magic.out
6 6	1
1	2
1 2	
2 3	
3 4	
4 2	
4 5	
3 6	
2	
5 6	
6 6	