

Problem A. Rectangle and Squares

Input file: **standard input**
Output file: **standard output**
Time limit: 2 seconds
Memory limit: 256 megabytes

Elijah came to visit his friend Phil and noticed that Phil owns an incredibly beautiful rectangle with sides A and B . Elijah realized that he had always dreamed about a rectangle of such area.

After returning to his home Elijah has found out that he has a lot of squares of size $C \times C$. He would like to construct a rectangle out of such squares that would have area as close as possible to the area of the Phil's rectangle. Namely, he wants to minimize the absolute difference between Phil's rectangle's area and his rectangle's area.

Elijah would put squares with sides parallel to each other without gaps and overlapping. He would use at least one square.

For example, if Phil has 4×5 rectangle and Elijah has 3×3 squares, the rectangle with the closest area to Phil's one that Elijah can construct is 3×6 .

Input

Input data contains several test cases. The first line contains the number of test cases t ($1 \leq t \leq 10000$).

Each test case is described with three integers in one line: A , B and C ($1 \leq A, B, C \leq 10^9$).

Output

For each test print one line: the area of the rectangle that Elijah would construct out of his squares, that has the closest possible area to the Phil's rectangle.

If there are several possible optimal answers, output any one.

Example

standard input	standard output
3	18
4 5 3	6
2 3 1	25
6 4 5	

Problem B. Zeroes and Ones

Input file: **standard input**
Output file: **standard output**
Time limit: 2 seconds
Memory limit: 256 megabytes

Peter hasn't submitted his homework for today's Computer Science lesson, so he is punished and given an additional task. The teacher has written two strings of the same length at the blackboard and asks Peter to make them equal using operations of only one type: take two adjacent characters of one of the strings and invert them both. Inversion transforms 0 to 1 and 1 to 0.

To make the problem even harder, Peter must use the minimal number of inversion operations.

Help Peter to complete his difficult task.

For example, if the two strings were "0101" and "1111" one way to make them equal is to invert two characters in the middle of the first string to get "0011" and "1111" and then invert two first characters of the second string to get "0011" and "0011". Note that there are other ways to complete the task with two operations in this example.

Input

Input data contains multiple test cases. The first line contains the number of test cases t ($1 \leq t \leq 100$).

Each test case is described in the following way. The first line contains n ($1 \leq n \leq 10^5$) — the length of the strings written by the teacher. The second and the third lines contain the strings themselves — two strings of length n containing only 0-s and 1-s.

The sum of values of n for all test cases in one input data doesn't exceed 10^5 .

Output

For each test case print one line: the minimal required number of inversion operations to make the strings equal, or -1 if it is impossible to make the strings equal.

Example

standard input	standard output
2	2
4	-1
0101	
1111	
3	
011	
111	

Problem C. New Track

Input file: **standard input**
Output file: **standard output**
Time limit: 2 seconds
Memory limit: 256 megabytes

Track designer Ger Kilke is going to create a new track for Formula-Y race. Ger is planning to use his favorite way to design the track. He introduced the Cartesian coordinates at the plane where the track will be built, X axis goes from left to right, Y axis goes from bottom to top. The track must satisfy the following conditions:

- Track must be a polyline composed of n segments parallel to coordinate axes.
- One of the polyline ends is a track start, the other one is a track finish, they must be at different points.
- Ends of track segments must be in points with positive integer coordinates not exceeding 3000.
- All segments must have a non-zero length.
- When traversing the track from start to finish, each time one proceeds to the next segment, he must turn 90 degrees clockwise.
- The end of each segment must not belong to the another segment, except adjacent segments, which share a common end. In particular, no two segments must have a common part with non-zero length.

Ger would like to make a track of n segments such that it had exactly k intersections. Mathematicians have told him that it is possible only if $0 \leq k \leq n_2(n_2 - 1)/2$, where n_2 is equal to $n/2$ rounded down. So Ger has chosen the value of k that satisfies this condition. Help Ger to design the track.

Input

Input data contains multiple test cases. The first line contains t — the number of tests.

Each of the following t lines contains two positive inegers: n and k ($1 \leq n \leq 1000$, $0 \leq k \leq n_2(n_2 - 1)/2$, where n_2 is equal to $n/2$ rounded down) — the number of segments of the track and the number of desired intersections.

The sum of values of n in one input data doesn't exceed 10^4 .

Output

Print the track description for each of t tests.

Each description must contain $n + 1$ lines, the i -th of these lines must contain coordinates of the i -th vertex of the polyline. List vertices from the start to the finish. Coordinates must be positive integers not exceeding 3000.

It is guaranteed that such polyline always exists for the given constraints.

Example

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

Problem D. Tree

Input file: **standard input**
 Output file: **standard output**
 Time limit: 2 seconds
 Memory limit: 256 megabytes

Nick has got a rooted tree with n vertices as his birthday present.

Recall that a rooted tree is a connected undirected graph, one vertex is chosen as a root. For each vertex the closest to the root neighbor is called the parent of this vertex, other neighbors are its children. The root has no parent. Nick's tree has vertices numbered from 1 to n , vertex with number 1 is the tree root.

Nick has decided to play with his tree. He chooses some vertex u and removes it, connecting the vertices that were its children to the former parent vertex of u . The root is never removed. But it is not interesting to just remove the vertices, so sometimes he asks for the length of the path from one vertex to another in the current tree. Answer to Nick's questions.

Input

The first line of the input contains integer n ($1 \leq n \leq 100\,000$) — the number of vertices in the initial tree. The second line contains $n - 1$ integers p_i ($1 \leq p_i \leq n$) — parents of vertices $2, 3, \dots, n$.

The third line contains integer q ($1 \leq q \leq 100\,000$) — the number of queries. Each of the following q lines contains queries. Each query starts with a query type — an integer equal either to 1 or to 2. If the query type is 1, it is followed by a and b ($1 \leq a, b \leq n$) — the numbers of vertices the distance between which must be found. If the query type is 2, one integer v ($1 \leq v \leq n$) follows — the number of the vertex to remove.

It is guaranteed that the initial tree is given correctly, the vertices in queries are not removed yet, and the root is never removed.

Output

For each query of type 2 print the distance between the corresponding vertices, one per line.

Example

standard input	standard output
8	4
1 5 2 1 2 5 5	2
7	3
2 4	2
1 7 6	
2 5	
1 3 8	
1 8 6	
2 2	
1 8 6	

Problem E. Barbarians

Input file: **standard input**
 Output file: **standard output**
 Time limit: 4 seconds
 Memory limit: 256 megabytes

Life on an island could be dangerous! Recently the country, which territory consists of n islands, was attacked by barbarians. The islands of the country were connected by the bridges in such way that for any two islands there was exactly one way to get from one island to another without visiting any island twice.

The barbarians understood that the bridges were the weak point of the country. So they decided to destroy them one after another. The people of the country got really angry. For the i -th island the initial anger of its people was a_i .

Let us describe how the anger changes when the bridge is destroyed. Consider an island x , let the number of islands that the people from x could reach before some bridge was destroyed be a , and the number of such islands after it was destroyed be b . Then the anger of people at x would be multiplied by $a - b + 1$.

You must develop the system to calculate the current anger of people. You will get several requests, each request contains two integers u and v . Request with integers u and v means that the bridge between $u + res$ and $v + res$ was destroyed, where res is the result of the previous query. For the first query consider $res = 0$.

Let the anger of people at island i after the bridge from the query was destroyed be b_i . Set res to be equal to the sum of all b_i for i from 1 to n , taken modulo $10^9 + 7$, and print this value.

To better understand the queries format, consider the example below.

Initially, the anger of people is

1 2 3 4 5

After removing the bridge between 1 and 3, the anger of people at islands 1 and 2 was multiplied by 4, and anger of people at other islands — by 3. So the new angriness values are

4 8 9 12 15

The sum of anger values is 48, we assign it to res , print it, and use it to find out that the bridge in the second query was between islands $(-47) + 48 = 1$ and $(-46) + 48 = 2$. The new values of angriness after this destruction are

8 16 9 12 15

The sum of anger values is 60. The next bridge to destroy was between islands 3 and 5. The new values of anger are

8 16 18 24 45

The sum of anger values is 111. After destroying the last bridge the anger values are

8 16 36 48 45

The sum is 153.

Input

The first line contains integer n ($2 \leq n \leq 2 \cdot 10^5$) — the number of islands.

The second line contains n integers a_i ($1 \leq a_i \leq 10^9 + 6$) — the initial angriness of people.

The following $n - 1$ lines contain two integers u_i, v_i each ($1 \leq u_i, v_i \leq n$) and describe bridges. It is guaranteed that there is exactly one way to get from any island to any other.

The following $n - 1$ lines describe queries. See description in the problem statement to better understand the format of queries. It is guaranteed that queries specify bridges that exist and were not yet destroyed.

Output

For each query print *res* value, one per line in the order of queries.

Example

standard input	standard output
5	48
1 2 3 4 5	60
1 2	111
1 3	153
3 4	
3 5	
3 1	
-47 -46	
-57 -55	
-108 -107	