

Driving License Exam

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 512 megabytes

Misha is a beginner driver. After successfully passing the theoretical part of the driving exam, he had little left to do — pass the actual driving exam. However, it turned out to be not so simple, as Misha does not know how to drive on ice.

The track where the exam takes place is located in a field and consists of n intersections, with a road of length d_i between the i -th and $(i + 1)$ -th intersections for i from 1 to $n - 1$. Initially, there are w_i units of ice at the i -th intersection.

For the exam, the inspectors choose a continuous subsegment $[l, r]$ of the track: all intersections with numbers from l to r inclusive and all the roads between them. The track for the exam must be circular, so after choosing l and r , the inspectors also connect the intersections with numbers l and r with a temporary road of length x .

The inspectors do not want Misha to pass the exam, so all the roads of the exam track must be covered with ice. To do this, it is necessary to move the ice from the intersections that the road connects; one unit of ice can cover one unit of the road. Hence, the total amount of ice moved onto the road must be not less than its length. Of course, no more ice can be moved onto two adjacent roads from each intersection than is present there.

Unfortunately, the current amount of ice may not be enough to cover all the roads, so the inspectors are interested in the minimum total amount of ice that needs to be added to the intersections to make it possible to cover all the roads and prevent Misha from passing the exam.

You will need to process three types of queries: change the amount of ice at a certain intersection, change the length of a certain road, and solve the problem for a subsegment of the track.

Input

The first line contains two integers n and q — the number of intersections and queries, respectively ($2 \leq n \leq 2 \cdot 10^5, 1 \leq q \leq 2 \cdot 10^5$).

The second line contains n integers w_i — the initial amount of ice at each intersection ($1 \leq w_i \leq 10^9$).

The third line contains $n - 1$ integers d_i — the initial length of each road between adjacent intersections ($1 \leq d_i \leq 10^9$).

The next q lines describe the queries:

- $1 \ p \ x$ ($1 \leq p \leq n, 1 \leq x \leq 10^9$) — perform the assignment $w_p := x$;
- $2 \ p \ x$ ($1 \leq p \leq n - 1, 1 \leq x \leq 10^9$) — perform the assignment $d_p := x$;
- $3 \ l \ r \ x$ ($1 \leq l < r \leq n, 1 \leq x \leq 10^9$) — determine the minimum amount of ice that needs to be added to the intersections for the track formed by the subsegment of intersections from l to r inclusive, adding temporary road of length x closing the cycle. Note that the third query does not change the amount of ice at the intersections; you only need to answer how much ice needs to be added to the intersections to cover all the roads with ice. After each query, the inspectors dismantle the temporary road, so it does not need to be considered in other queries.

It is guaranteed that there is at least one query of the third type.

Output

For each query of the third type, output the minimum total amount of ice that needs to be added to the intersections to cover the entire circular track with ice.

Example

standard input	standard output
6 7	9
5 9 5 1 9 5	0
4 8 10 4 5	1
3 1 6 12	6
3 4 6 5	
2 4 1	
1 6 3	
3 4 6 5	
1 2 3	
3 2 3 6	

Note

The figure shows illustrations for the example. Dotted lines indicate temporary roads closing the cycle. The current and added amounts of ice are indicated at the intersections, and the amount of ice moved from the intersections is indicated on the roads.

