A. Simulating Move to Front

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	256 megabytes

Start with a list $[0, 1, 2, \ldots, n-1]$. Now you get a sequence of m **move to front** requests, where `mtf`(i) moves element i to the front of the list. When it does the move, your program should output the index of the element being moved. (Zero-based indexing is used, so the front of the list has index 0.)

In this way, a sequence of m numbers in the range $[0, n-1]$ is transformed into another such sequence. This transformation (and its inverse) is useful in implementing a certain data compression algorithm.

So, for example, say $n = 5$. The initial list is $[0, 1, 2, 3, 4]$. If `mtf(3)` is executed, then the list becomes $[3, 0, 1, 2, 4]$, and 3 is output. If the next request is 3, then the list stays the same and 0 is output. Finally if the next request is `mtf(4)` then the list becomes $[4, 3, 0, 1, 2]$ and 4 is output.

Input

The first line contains blank-separated numbers n and m with $1 \le n \le 10^6$ and $1 \le m \le 5 \times 10^5$. The second line consists of the numbers $p_1, p_2, \ldots, p_m$, each of which is in the range $[0, n-1]$. These are the items to which the move-to-front operation is applied.

Output

The output consists of one line containing m space-separated numbers. The ith of these is the index of the item p_i when it is requested.

Examples

standard input	standard output

`5 3`
`3 3 4`
	`3 0 4`

`6 8`
`5 0 4 0 3 0 2 0`
	`5 1 5 1 5 1 5 1`

B. Mass Change Queries

3 seconds, 512 megabytes

You are given an array a consisting of n integers. You have to process q queries to this array; each query is given as four numbers l, r, x and y, denoting that for every i such that $l \le i \le r$ and $a_i = x$ you have to set a_i equal to y.

Print the array after all queries are processed.

Input

The first line contains one integer n ($1 \le n \le 200000$) — the size of array a.

The second line contains n integers a_1, a_2, ..., a_n ($1 \le a_i \le 100$) — the elements of array a.

The third line contains one integer q ($1 \le q \le 200000$) — the number of queries you have to process.

Then q lines follow. i-th line contains four integers l, r, x and y denoting i-th query ($1 \le l \le r \le n$, $1 \le x, y \le 100$).

Output

Print n integers — elements of array a after all changes are made.

input

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

output

```
5 2 5 4 5
```

C. Drazil and Park

2 seconds, 512 megabytes

Drazil is a monkey. He lives in a circular park. There are n trees around the park. The distance between the i-th tree and ($i + 1$)-st trees is d_i, the distance between the n-th tree and the first tree is d_n. The height of the i-th tree is h_i.

Drazil starts each day with the *morning run*. The morning run consists of the following steps:

- Drazil chooses two different trees
- He starts with climbing up the first tree
- Then he climbs down the first tree, runs around the park (in one of two possible directions) to the second tree, and climbs on it
- Then he finally climbs down the second tree.

But there are always children playing around some consecutive trees. Drazil can't stand children, so he can't choose the trees close to children. He even can't stay close to those trees.

If the two trees Drazil chooses are x-th and y-th, we can estimate the energy the *morning run* takes to him as $2(h_x + h_y) + dist(x, y)$. Since there are children on exactly one of two arcs connecting x and y, the distance $dist(x, y)$ between trees x and y is uniquely defined.

Now, you know that on the i-th day children play between a_i-th tree and b_i-th tree. More formally, if $a_i \le b_i$, children play around the trees with indices from range $[a_i, b_i]$, otherwise they play around the trees with indices from $[a_i, n] \cup [1, b_i]$.

Please help Drazil to determine which two trees he should choose in order to consume the most energy (since he wants to become fit and cool-looking monkey) and report the resulting amount of energy for each day.

Input

The first line contains two integer n and m ($3 \le n \le 10^5$, $1 \le m \le 10^5$), denoting number of trees and number of days, respectively.

The second line contains n integers $d_1, d_2, ..., d_n$ ($1 \le d_i \le 10^9$), the distances between consecutive trees.

The third line contains n integers $h_1, h_2, ..., h_n$ ($1 \le h_i \le 10^9$), the heights of trees.

Each of following m lines contains two integers a_i and b_i ($1 \le a_i, b_i \le n$) describing each new day. There are always at least two different trees Drazil can choose that are not affected by children.

Output

For each day print the answer in a separate line.

input

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

output

```
12
16
18
```

input

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

output

```
17
22
11
```

Problem D. Journey to the “The World’s Start”

Input file:	`journey.in`
Output file:	`journey.out`
Time limit:	2 seconds
Memory limit:	256 megabytes

Jerry Prince is the fourth grade student and he goes to New-Lodnon to visit the most popular amusement park “The World’s Start”.

An airport he arrives at is next to the first stop of the metro line. This line has n stops and “The World’s Start” is on the last of them. The metro of New-Lodnon is pretty fast so you may assume that you can get from a stop to the next one in just one minute.

Jerry needs a travel card to use the metro. Each travel card has a range r and a price p. With a travel card of range r Jerry may travel no more than r stops at once. Therefore, if Jerry enters metro at the stop i he should exit on one of the stops from $i - r$ to $i + r$ inclusive. It takes d_i minutes to exit and reenter metro at i-th stop. There is no time required to enter the first stop or exit the last one.

Jerry is not very rich but he has some spare time, so he decided to buy the cheapest travel card that will allow him to travel from the first metro stop to the last one in no more than t minutes.

Input

The first line of the input file contains two integers n and t — the number of stops and the maximum possible time ($2 \le n \le 50\,000$; $n - 1 \le t \le 10^9$).

The second line contains $n - 1$ integers p_r — the prices of travel cards with range $r = 1 \ldots n - 1$ ($1 \le p_r \le 100\,000$)

The third line contains $n - 2$ integers d_i — the number of minutes required to reenter metro at stop $i = 2 \ldots n - 1$ ($1 \le d_i \le 10^5$).

Output

Output a single integer p — the lowest possible price of one travel card that allows Jerry to travel from the first to the last stop in no more than t minutes.

Example

journey.in	journey.out

4 4
1 2 3
1 4
	2

Problem E. Hiring

Input file:	`standard input`
Output file:	`standard output`
Time limit:	4 seconds
Memory limit:	512 megabytes

The head of human resources department decided to hire a new employee. He created a test exercise for candidates which should be accomplished in at most m working days. Each candidate has to pass this test exercise. During the j-th day a candidate is allowed to be in the office for at most t_j units of time.

Overall, n candidates decided to apply for the job and sent out their resumes. Based on data received the head has defined two parameters describing every candidate: d_i and r_i. The parameter d_i is the time to get prepared for work which the i-th candidate spends each morning. This time doesn't depend on day. The parameter r_i is the total working time needed for the i-th candidate to accomplish the whole test exercise.

Thus the time spent in the office in the j-th day consists of d_i units of time to get prepared and some units of time to proceed with the exercise. A candidate can skip entire working day and do not come to the office. Obviously in this case he doesn't spend d_i units of time to prepare.

To complete the exercise a candidate should spend exactly r_i units of time working on the exercise (time to prepare is not counted here).

Find out for each candidate what is the earliest possible day when he can fully accomplish the test exercise. It is allowed to skip working days, but if candidate works during a day then he must spend d_i units of time to prepare for work before he starts progressing on the exercise.

Input

The first line contains two integer numbers n, m ($1 \le n, m \le 2 \cdot 10^5$) — the number of candidates and the maximum number of working days to do the test exercise.

The second line contains m integer numbers $t_1, t_2, \ldots, t_m$ ($1 \le t_j \le 10^6$) — the durations of working days in time units.

The following n lines contain two integers each: d_i, r_i ($0 \le d_i \le 10^6$, $1 \le r_i \le 10^6$) — how much time in the beginning of a day is required for i-th candidate before he starts his work on the test exercise and how much time it is needed for him to accomplish this task.

Output

Output a sequence of n integer numbers $b_1, b_2, \ldots, b_n$, where b_i is the earliest day when the i-th candidate can finish the test exercise.

In case the i-th candidate cannot finish the test exercise in m days output $b_i = 0$.

Days in this problem are numbered from 1 to m in the order they are given in the input.

Examples

standard input	standard output

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