

Birdwatchers

Time limit: 10 s Memory limit: 512 MiB

The Birdwatchers' Society of San Serriffe has a curiously bloated and ever-changing internal structure. The Society consists of n *chapters*, and each member of the Society belongs to exactly one chapter. The chapters are numbered from 1 to n and the i 'th chapter has m_i members. Thus the Society has a total of $M = m_1 + m_2 + \dots + m_n$ members.

Each chapter is led by one of its members, who in this role is called the *officer* of the chapter. The officers are numbered the same as the chapters, so that (for each $i = 1, \dots, n$) officer i is the one in charge of chapter i .

Moreover, the officers are organized hierarchically through a system of mentorship: each officer except one has a *mentor*, who is the officer of some other chapter. The only officer without a mentor is the *President* of the Society. If officer a is the mentor of officer b , we also say that officer b is a *disciple* of officer a . No officer is, directly or indirectly, a mentor of themselves; thus, by following the sequence from an officer to their mentor, their mentor's mentor, and so on, we eventually always reach the President.

We define the *influence* of an officer as the sum of the number of members in his chapter and of the influences of all his disciples (if he has any). It will be easily seen that the officer with the highest influence is the President, whose influence always equals M . An officer is called a *senior officer* if his influence is $\geq M/2$.

The Bylaws of the Society specify that the senior officer with the lowest influence (amongst all the senior officers) shall act as the *Treasurer* of the Society.

From time to time, an officer (other than the President) may *change his allegiance*, so that he is thenceforth the disciple of a different mentor than before (provided that his new mentor is not one of his disciples, or his disciples' disciples, etc.). Because of this, it can happen that the influence of some officers changes and the role of Treasurer falls to a different officer than before.

Task

Write a program that reads the description of the initial state of the Society and a sequence of changes of allegiance. Your program must print who the Treasurer is in the initial state of the Society as well as after each change of allegiance.

Input

The first line contains two integers, n and q , separated by a space; n is the number of chapters, q is the number of changes of allegiance.

The next n lines describe the initial state of the Society. The i 'th of these lines contains two integers, s_i and m_i , separated by a space; s_i is the mentor of officer i (i.e. of the officer in charge of chapter i), while m_i is the number of members of chapter i . The value $s_i = 0$ indicates that officer i is the President of the Society and thus has no mentor.

The remaining q lines describe the changes of allegiance. The j 'th of these lines contains two integers, $\hat{x}_j$ and $\hat{z}_j$, separated by a space. The meaning of these integers is as follows. Let us denote by t_j (for $j = 0, \dots, q$) the Treasurer after the first j changes of allegiance (thus t_0 is the initial Treasurer before the first change of allegiance). Then the j 'th change of allegiance consists of officer z_j becoming the new mentor of officer x_j ,

where $x_j = 1 + ((t_{j-1} + \hat{x}_j) \bmod n)$ and $z_j = 1 + ((t_{j-1} + \hat{z}_j) \bmod n)$. The purpose of this representation of the values x_j and z_j is to force your program to process the changes of allegiance in the order in which they appear.

The changes of allegiance in the input data will always be valid, i.e. z_j will not be equal to x_j , nor will z_j be a disciple of x_j , a disciple of a disciple of x_j , etc. It is, however, possible that z_j was already the mentor of x_j just before the j 'th change (so that nothing actually changes at that point).

Note that if your program calculates the wrong result t_j at some point, it will decode the subsequent inputs $\hat{x}_{j+1}, \hat{z}_{j+1}$, etc., incorrectly as well, and might crash with an RTE (runtime error) verdict instead of a WA (wrong answer) verdict because the incorrectly decoded inputs might be invalid (e.g. it might erroneously obtain a z_{j+1} which is a disciple of x_{j+1}).

Constraints

- $1 \leq n \leq 1\,000\,000$
- $1 \leq q \leq 30\,000$
- $1 \leq m_i$ for each $i = 1, \dots, n$
- $m_1 + m_2 + \dots + m_n \leq 10^9$
- $1 \leq \hat{x}_j \leq n$ and $1 \leq \hat{z}_j \leq n$ for each $j = 1, \dots, q$.

Subtasks

- Subtask 1 (15 points): $n \leq 100$
- Subtask 2 (10 points): $n \leq 1000$
- Subtask 3 (50 points): $n \leq 300\,000$
- Subtask 4 (25 points): No additional constraints.

Output

Print the numbers $t_0, t_1, \dots, t_q$, each on its own line, where t_j is the Treasurer after the first j changes of allegiance. Naturally, each t_j must be an integer from the range $1 \leq t_j \leq n$.

Example

Input

7 2
0 1
1 3
1 3
2 3
2 1
5 2
5 1
3 7
2 7

Output

2
2
3

Comment

Initially, officer 2 is the Treasurer (hence $t_0 = 2$). In the first change of allegiance, we read $\hat{x}_1 = 3$ and $\hat{z}_1 = 7$ and calculate $x_1 = 1 + ((2+3) \bmod 7) = 6$ and $z_1 = 1 + ((2+7) \bmod 7) = 3$; thus, officer 3 becomes the new mentor of officer 6; officer 2 remains Treasurer (hence $t_1 = 2$). In the second change of allegiance, we read $\hat{x}_2 = 2$ and $\hat{z}_2 = 7$ and calculate $x_2 = 1 + ((2+2) \bmod 7) = 5$ and $z_2 = 1 + ((2+7) \bmod 7) = 3$; thus, officer 3 becomes the new mentor of officer 5 and also becomes the new Treasurer (hence $t_2 = 3$).