TRAVEL

1s

In a far away country, there is a tourist named Thanh, who hopes to travel across the country on his bicycle. In this country, each city has an index, and there are n cities, numbered from 1 to n. A city either has no scenic spots that Thanh wants to visit, or has exactly 1 scenic spot that Thanh wants to visit.

Thanh has arranged a route going through all the attractions he wanted to visit: the 1st destination city is a_1, the i-th destination city is a_i, and finally, arriving at city a_n ends the trip. Thanh hopes to complete the trip exactly in month m ($m < n$), so he needs to plan his travel properly.

When Thanh arrives in a city, if the city has a tourist attraction that he wants to visit, he will receive 1 happy point, else Thanh will receive 1 sad point. A month is enough time for him to visit as many cities as he wants, but he also wants to rest a bit. Thanh always rests at the last city he visits that month (but if there is an attraction in this city, Thanh will always visit it before resting). Of course, Thanh hopes to have a certain travel plan every month, even if there are no attractions he wants to visit in the destination city during the month; in other words, he will visit at least one new city every month.

Thanh needs to come up with a travelling/resting schedule, i.e, a string $x_1, x_2, \ldots, x_m$, where x_i is the index number of the city at where Thanh rests in month i. Since he has to finish his trip after the last month, $x_m = a_n$. For example: n = 5, m = 3, $(a_1, a_2, a_3, a_4, a_5) = (3, 2, 4, 1, 5)$, $(x_1, x_2, x_3) = (2, 1, 5)$, meaning: In month 1 he goes to cities 3 and 2, and rests at city 2; in month 2 he goes to city 4 and city 1, and rests at city 1; in month 3 he goes to city 5 and rests at city 5. There are many such plans.

For a schedule K, let the absolute value of the difference between the happy value and the sad value obtained during the trip of month i be d_i. The largest value in the sequence $d_1, d_2, \ldots, d_m$ is called C_K.

Thanh hopes that the C_K of the selected schedule is the smallest among all the schedules.

In fact, there can be many schedules in which C_K is smallest, so he wants to find the schedule with the smallest lexicographical order.

Pro tip: Comparing the lexical order of two strings is comparing the first different number, such as (1,2,3,4) < (1,2,4,3).

INPUT

The first line contains two positive integers n, m, indicating the number of cities and the number of months spent travelling ($n \leq 5\times10^5$, $m \leq 2\times10^5$).

Each of the next n lines contains 2 integers A_i and B_i which are the ordinal numbers of the city where Thanh visited i and B_i is 0 or 1. If $B_i = 0$ (1), it means that A_i city does not have (has) the tourist attractions that Thanh likes. The values Ai are distinct and $1 \leq A_i \leq N$, meaning $\{A_i\}$ is a permutation of $\{1,\ldots,N\}$.

OUTPUT

Print out on one line m positive integers $x_1, x_2, \ldots, x_m$, which means the route corresponding to Thanh's travel plan.

Sample Input	Sample Output
8 3 2 0 3 1 4 1 1 0 5 0 6 1 7 1 8 0	1 6 8

EXPLAIN

2 happy points and 2 sad points in month 1, 1 happy point and 1 sad point in month 2, and 1 happy point and 1 sad point in month 3. The maximum value of the difference between sad value and happy value at month 3 is 0, which is the minimum value for all schedules. Possible solutions are:

1 6 8

3 6 8

3 1 8

Among them, 1 6 8 is the smallest lexicographic order.