

Delivering the Bread

Problem ID: deliveringbread

Kiki has taken the odd job of delivering bread across the entire globe!

She is currently passing through a group of cities on the equator. The equator has a total distance of D , with positions labeled 0 to $D - 1$. Kiki only moves one step in the positive direction at a time, except when she is at position $D - 1$, where she will instead go back to position 0 .

In these D positions, there are N customers. Each customer can be described by its position on the equator, d_i , and the type of bread they want, which is a number k_i between 0 and $K - 1$. It is guaranteed that at least one customer wants each type of bread. Kiki can visit multiple customers at the same location without traveling any distance.



Kiki has bread of every single type, but as the different bread types are stacked on top of each other, she can only deliver the type of bread on the top of her stack. Moreover, Kiki can only remove bread from the top of her stack by delivering it to customers.

The type of the next bread on the stack is always 1 more than the last type of bread, unless the last type of bread is of type $K - 1$, in which the following bread type will be of type 0 . More formally, if the top of Kiki's bread stack is of type D , the next bread will be of type $(D + 1) \bmod K$. Kiki can deliver multiple pieces of bread at a location, without moving, if multiple suitable bread customers at that location exist.

Kiki currently is considering Q routes. In the q th query, she starts at some initial position d'_q , has some initial bread type on the top of her stack k'_q , and wants to deliver a total of n'_q pieces of bread.

Compute, for each query, how far Kiki has to travel in order to deliver the amount of bread that she desires.

Input

The first line contains 4 space-separated integers, N ($1 \leq N \leq 10^5$), K ($1 \leq K \leq N$), D ($1 \leq D \leq 10^9$), and Q ($1 \leq Q \leq 10^5$). N represents the number of customers, K represents the number of bread types, D represents the number of positions on the equator, and Q represents the number of routes.

The second line contains N space-separated integers d_1, d_2, \dots, d_N ($0 \leq d_i \leq D - 1$) detailing the positions of the customers. The third line contains N space-separated integers k_1, k_2, \dots, k_N ($0 \leq k_i \leq K - 1$) detailing what type of bread each customer wants.

Then follow Q lines, with each line describing one route. Each of these lines contains three space-separated integers containing the starting position of Kiki on this route, d'_q ($0 \leq d'_q \leq D - 1$), the type of bread on the top of her stack when beginning this route, k'_q ($0 \leq k'_q \leq K - 1$), and the total number of pieces of bread she wishes to deliver on this route, n'_q ($0 \leq n'_q \leq K$).

Output

For each query, output the distance Kiki has to travel. Output the answer to each query on a separate line, in the order of the queries given.

Sample Input 1	Sample Output 1
5 3 10 3	11
1 3 5 7 9	3
2 1 2 0 1	11
6 2 2	
8 2 1	
0 0 3	