

E. Camping Groups

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

A club wants to take its members camping. In order to organize the event better the club directors decided to partition the members into several groups.

Club member i has a responsibility value r_i and an age value a_i . A *group* is a non-empty subset of club members with one member known as group leader. A group leader should be one of the most responsible members of the group (his responsibility value is not less than responsibility of any other group member) and his age absolute difference with any other group member should not exceed k .

Some club members are friends and want to be in the same group. They also like their group to be as large as possible. Now you should write a program that answers a series of questions like "What's the largest size of a group containing club member x and club member y ?". It's possible for x or y to be the group leader.

Input

The first line contains two integers n and k ($2 \leq n \leq 10^5$, $0 \leq k \leq 10^9$) — the number of club members and the age restriction for one group.

The next line contains integer numbers r_1, r_2, \dots, r_n ($1 \leq r_i \leq 10^9$) separated by space: r_i denotes the i -th club member's responsibility. In the same way there are integers a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$) in the third line: a_i denotes the i -th club member's age.

The next line contains an integer q denoting the number of questions that you should answer ($1 \leq q \leq 10^5$). The next q lines describe the questions. Each line contains two space-separated integers x_i and y_i ($1 \leq x_i, y_i \leq n$, $x_i \neq y_i$) — the indices of the club members that should end up in the same group.

Output

For each question print the maximum size of the group in a line. If making such a group is impossible print -1 instead.

Examples

input
5 1 1 5 4 1 2 4 4 3 2 2 4 5 3 2 3 2 5 4 1
output
4 3 -1 4

Note

In the first query the largest group with members 3 and 5 is $\{1, 3, 4, 5\}$ where member 3 is the leader.

In the second query member 2 should be the leader so the group will be $\{1, 2, 3\}$.

In the third query the leader of the group should have age 3 so the only leader can be member 3, who is less responsible than member 2. So making a group is impossible.

The group for the fourth query is the same as first query.