A. Bear and Displayed Friends

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

Limak is a little polar bear. He loves connecting with other bears via social networks. He has n friends and his relation with the i-th of them is described by a unique integer t_i. The bigger this value is, the better the friendship is. No two friends have the same value t_i.

Spring is starting and the Winter sleep is over for bears. Limak has just woken up and logged in. All his friends still sleep and thus none of them is online. Some (maybe all) of them will appear online in the next hours, one at a time.

The system displays friends who are online. On the screen there is space to display at most k friends. If there are more than k friends online then the system displays only k best of them — those with biggest t_i.

Your task is to handle queries of two types:

- "1 id" — Friend id becomes online. It's guaranteed that he wasn't online before.
- "2 id" — Check whether friend id is displayed by the system. Print "YES" or "NO" in a separate line.

Are you able to help Limak and answer all queries of the second type?

Input

The first line contains three integers n, k and q ($1 \le n, q \le 150\,000$, $1 \le k \le min(6, n)$) — the number of friends, the maximum number of displayed online friends and the number of queries, respectively.

The second line contains n integers $t_1, t_2, \ldots, t_n$ ($1 \le t_i \le 10^9$) where t_i describes how good is Limak's relation with the i-th friend.

The i-th of the following q lines contains two integers $type_i$ and id_i ($1 \le type_i \le 2$, $1 \le id_i \le n$) — the i-th query. If $type_i = 1$ then a friend id_i becomes online. If $type_i = 2$ then you should check whether a friend id_i is displayed.

It's guaranteed that no two queries of the first type will have the same id_i becuase one friend can't become online twice. Also, it's guaranteed that at least one query will be of the second type ($type_i = 2$) so the output won't be empty.

Output

For each query of the second type print one line with the answer — "YES" (without quotes) if the given friend is displayed and "NO" (without quotes) otherwise.

Examples

input
4 2 8 300 950 500 200 1 3 2 4 2 3 1 1 1 2 2 1 2 2 2 3
output
NO YES NO

VK Cup 2016 - Round 1

Finished

→ Virtual participation

Virtual contest is a way to take part in past contest, as close as possible to participation on time. It is supported only ACM-ICPC mode for virtual contests. If you've seen these problems, a virtual contest is not for you - solve these problems in the archive. If you just want to solve some problem from a contest, a virtual contest is not for you - solve this problem in the archive. Never use someone else's code, read the tutorials or communicate with other person during a virtual contest.

Start virtual contest

→ Problem tags

implementation

No tag edit access

→ Contest materials

- Announcement
- Tutorial

```
YES
YES
```

input

```
6 3 9
50 20 51 17 99 24
1 3
1 4
1 5
1 2
2 4
2 2
1 1
2 4
2 3
```

output

```
NO
YES
NO
YES
```

Note

In the first sample, Limak has 4 friends who all sleep initially. At first, the system displays nobody because nobody is online. There are the following 8 queries:

1. "1 3" — Friend 3 becomes online.
2. "2 4" — We should check if friend 4 is displayed. He isn't even online and thus we print "NO".
3. "2 3" — We should check if friend 3 is displayed. Right now he is the only friend online and the system displays him. We should print "YES".
4. "1 1" — Friend 1 becomes online. The system now displays both friend 1 and friend 3.
5. "1 2" — Friend 2 becomes online. There are 3 friends online now but we were given $k = 2$ so only two friends can be displayed. Limak has worse relation with friend 1 than with other two online friends ($t_1 < t_2, t_3$) so friend 1 won't be displayed
6. "2 1" — Print "NO".
7. "2 2" — Print "YES".
8. "2 3" — Print "YES".