G. Modular Sorting

time limit per test: 5 seconds

memory limit per test: 256 megabytes

You are given an integer m ($2 \leq m \leq 5 \cdot 10^5$) and an array a consisting of nonnegative integers smaller than m.

Answer queries of the following form:

- $1\ i\ x$: assign $a_i := x$
- $2\ k$: in one operation, you may choose an element a_i and assign $a_i := (a_i + k) \pmod m$* — determine if there exists some sequence of (possibly zero) operations to make a nondecreasing†.

Note that instances of query 2 are independent; that is, no actual operations are taking place. Instances of query 1 are persistent.

---
*$a \pmod m$ is defined as the unique integer b such that $0 \leq b < m$ and $a - b$ is an integer multiple of m.

†An array a of size n is called nondecreasing if and only if $a_i \leq a_{i+1}$ for all $1 \leq i < n$.

Input

The first line contains an integer t ($1 \leq t \leq 10^4$) — the number of test cases.

The first line of each test case contains three integers, n, m, and q ($2 \leq n \leq 10^5$, $2 \leq m \leq 5 \cdot 10^5$, $1 \leq q \leq 10^5$) — the size of the array a, the integer m, and the number of queries.

The second line of each test case contains n integers, $a_1, a_2, \ldots, a_n$ ($0 \leq a_i < m$).

Then follows q lines. Each line is of one of the following forms:

- $1\ i\ x$ ($1 \leq i \leq n, 0 \leq x < m$)
- $2\ k$ ($1 \leq k < m$)

It is guaranteed that the sum of n and the sum of q over all test cases each do not exceed 10^5.
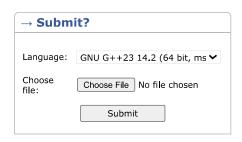
Output

For each instance of query 2, output on a single line "YES" if there exists some sequence of (possibly zero) operations to make a nondecreasing, and "NO" otherwise.

You can output the answer in any case (upper or lower). For example, the strings "yEs", "yes", "Yes", and "YES" will be recognized as positive responses.

Example

input		Copy

```
2
7 6 6
4 5 2 2 4 1 0
2 4
1 4 5
2 4
2 3
1 7 2
2 3
8 8 3
0 1 2 3 4 5 6 7
2 4
1 3 4
2 4
```

```
output                                                    Copy
YES
NO
NO
YES
YES
NO
```

Note

In the first sample, the array is initially:

4	5	2	2	4	1	0

By applying the operation twice on a_1, twice on a_2, once on a_5, twice on a_6, and once on a_7, the array becomes:

0	1	2	2	2	3	4

which is in nondecreasing order.

After the second query, the array becomes:

4	5	2	5	4	1	0

and it can be shown that it is impossible to sort this with operations of the form $a_i := (a_i + 4) \pmod 6$, and it is also impossible to sort this with operations of the form $a_i := (a_i + 3) \pmod 6$.