

Max Partition

Input file: **standard input**
Output file: **standard output**
Time limit: 2.5 seconds
Memory limit: 256 megabytes

*After escaping from guns, Evirir the dragon
is too tired to write a flavor text this time.*

There is a positive integer M . There are two lists of non-negative integers A_0 and A_1 , and all integers in the lists are between 0 and M (inclusive).

You can do the following operation zero or more times:

- Choose a list A_i ($i = 0$ or 1) and choose an integer x in A_i . Remove (one occurrence of) x from A_i and insert (one occurrence of) $M - x$ into A_{1-i} (that is, the other list).

Additionally, after all operations, A_0 and A_1 must be nonempty. A list can become empty after an operation, but after **all** operations, both lists must be nonempty.

You wonder whether you can perform these operations such that the maximum value in each list is equal to a specific value. You also wonder whether the answer changes if you add or remove integers.

Process Q queries in order, where each query is in one of the following forms:

- Type 1 query: $1\ i\ x$
 - Add (one occurrence of) x to A_i .
- Type 2 query: $2\ i\ x$
 - Remove (one occurrence of) x from A_i .
- Type 3 query: $3\ c_0\ c_1$
 - Answer the question: Is it possible to make $\max(A_0) = c_0$ and $\max(A_1) = c_1$ after doing zero or more operations¹? Output YES or NO on a line.

Note: For Type 3 queries, you do not actually do any operations; you are only checking whether it is possible.

¹Here, $\max(A_i)$ means the largest integer in A_i .

Input

The first line of the input contains three space-separated integers N_0 , N_1 , and M ($0 \leq N_0, N_1 \leq 2 \cdot 10^5$, $N_0 + N_1 \leq 2 \cdot 10^5$, $0 \leq M \leq 10^9$), where N_0 and N_1 are the length of A_0 and A_1 , respectively.

The second line contains N_0 integers, the elements of A_0 . For each integer x in A_0 , $0 \leq x \leq M$. This is a blank line if $N_0 = 0$.

The third line contains N_1 integers, the elements of A_1 . For each integer x in A_1 , $0 \leq x \leq M$. This is a blank line if $N_1 = 0$.

The fourth line contains an integer Q ($1 \leq Q \leq 2 \cdot 10^5$), the number of queries.

The next Q lines contain the queries in order. Each query consists of three space-separated integers and has one of the following forms:

- 1 $i x$ ($i = 0$ or 1 , $0 \leq x \leq m$)
- 2 $i x$ ($i = 0$ or 1 , $0 \leq x \leq m$)
 - It is guaranteed that A_i will contain x when this query is given.
- 3 $c_0 c_1$ ($0 \leq c_0, c_1 \leq m$)
 - It is guaranteed that the total number of integers in A_0 and A_1 is at least 2 when this query is given.

It is guaranteed that there is at least one Type 3 query.

Output

For each Type 3 query in the given order, if the answer is yes, output YES on a line. Otherwise, output NO on a line.

You may output each letter of YES and NO in any case. For example, yes and yEs will be treated like YES; no and No will be treated like NO.

Scoring

Subtask 1 (17 points): $N_0, M, Q \leq 100$. $N_1 = 0$. All queries are Type 3 queries with $c_0 = M - c_1$.

Subtask 2 (9 points): $N_1 = 0$. All queries are Type 3 queries with $c_0 = M - c_1$.

Subtask 3 (14 points): $M \leq 2$. All queries are Type 3 queries.

Subtask 4 (28 points): $N_0 + N_1 \leq 2000$, $Q \leq 2000$. All queries are Type 3 queries.

Subtask 5 (21 points): All queries are Type 3 queries.

Subtask 6 (11 points): No additional constraints.

Examples

standard input	standard output
5 4 10 3 4 6 7 0 1 3 7 8 11 3 7 3 3 0 1 3 6 9 3 4 4 3 7 8 2 1 3 1 0 5 1 1 2 3 4 4 3 0 10 3 7 5	YES NO NO YES YES NO NO YES
10 0 8 2 2 3 3 3 1 5 5 5 6 4 3 2 6 3 6 2 3 5 3 3 3 5	YES NO YES YES
0 0 10 5 1 1 4 1 1 6 2 1 4 1 0 4 3 4 6	YES

Note

Sample 1

For the first query (3 7 3), we can do the following:

- Choose 8 from A_1 : Remove 8 from A_1 and insert $10 - 8 = 2$ into A_0 .
- Choose 7 from A_1 : Remove 7 from A_1 and insert $10 - 7 = 3$ into A_0 .

This results in $A_0 = [3, 4, 6, 7, 0, 2, 3]$ and $A_1 = [1, 3]$. The maximum element in A_0 and A_1 are 7 and 3 respectively as desired, so the answer is YES.

For the fourth query (3 4 4), we can operate on 6 and 7 from A_0 and 7 and 8 from A_1 . This results in $A_0 = [3, 4, 0, 3, 2]$ and $A_1 = [1, 3, 4, 3]$.

For the fifth query (3 7 8), the maximum elements in A_0 and A_1 are already 7 and 8 respectively, so no operation is needed.

After the sixth query (2 1 3), $A_1 = [1, 7, 8]$. After the 7th and 8th queries (1 0 5 and 1 1 2), $A_0 = [3, 4, 6, 7, 0, 5]$ and $A_1 = [1, 7, 8, 2]$.

Sample 2

This sample satisfies the constraints of Subtask 1 and 2. For the first query, we can remove one of the 2s in A_0 and insert $8 - 2 = 6$ into A_1 .

Sample 3

Note that both lists can be empty initially.