Problem D. Decisions

Input file:	`decisions.in`
Output file:	`decisions.out`
Time limit:	5 seconds
Memory limit:	512 mebibytes

Denny is preparing for a class trip. Denny has several items that he is planning to take to the trip. He has put them as a single row and numbered from 1 to n. For each item Denny knows its weight w_i. The knapsack Denny is going to take to the trip has capacity of c, so Denny can take several items with total weight at most c.

Unfortunately Denny is very bad in making decisions. He cannot decide which items would be more useful in a trip. Additionally he doesn't like carrying too much, so probably he is not going to fill his knapsack completely. To help himself choose, Denny has decided to set additional constraints about items to choose.

Now Denny generates various plans of constraints. Each plan is a triple (l_j, r_j, t_j) where $1 \le l_j \le r_j \le n$ and $0 \le t_j \le c$. To satisfy constraints plan Denny must choose several items with numbers between l_j and r_j, inclusive, with the total weight exactly equal to t_j. For each plan Denny immediately wants to know whether it is possible to satisfy it. Help him!

Input

The input file contains several test cases.

Each test case starts with an integer n on a line ($1 \le n \le 2000$). The second line of each test case contains n integers w_i ($1 \le w_i \le 100$). The third line contains the capacity of the knapsack c ($1 \le c \le 1000$). The fourth line contains q — the number of plans that Denny is going to propose ($1 \le q \le 300\,000$). Each of the following q lines describe plans.

The j-th plan is described by three integers: a_j, b_j, c_j ($0 \le a_j < n$, $1 \le b_j \le n$, $0 \le c_j \le c$). Let there be k plans in this test case before the j-th one, that can be satisfied. Use the following equation to get the plan: $l_j = (a_j + k) \bmod (n - b_j + 1) + 1$, $r_j = l_j + b_j - 1$, and $t_j = (c_j + k) \bmod (c + 1)$.

The last test case is followed by a line that contains a single 0, it must not be processed. The sum of n for all test cases in one input file doesn't exceed 2000. The sum of q for all test cases in one input file doesn't exceed 300 000.

Output

For each test case output a line of characters '`Y`' and '`N`'. For each plan print '`Y`' if it can be satisfied, or '`N`' if it cannot.

Examples

decisions.in	decisions.out
5 2 3 4 5 9 10 5 0 3 5 0 2 4 0 3 4 0 4 6 2 3 5 0	YNYYN

In the given example the plans are $(1, 3, 5)$, $(2, 3, 5)$, $(2, 4, 5)$, $(1, 4, 8)$, $(3, 5, 8)$.