Problem H. Harumachi Kaze

Input file:	*standard input*
Output file:	*standard output*
Time limit:	30 seconds
Memory limit:	256 mebibytes

This is an interactive problem.

You are given two arrays a and b of length n, consisting of non-negative integers.

There is a hidden permutation of integers from 0 to $2^{64}-1$: $p(0), p(1), \ldots, p(2^{64}-1)$. You only know that $p(0) = 0$.

Since p is a permutation, p^{-1} can also be defined: $p^{-1}(x) = y$ when $p(y) = x$.

Also, you are given an integer B. A positive integer x is called **cute** if and only if the two following conditions hold:

- The binary representation of x, when viewed as a string of length B, is a palindrome.
- If a bit in the binary representation of x is 1, this bit must either be in the first 6 bits or in the last 6 bits. For example, if $B = 14$, the two bits in the center must both be 0.

You have to support queries of two types.

The first type of query is to change an element in one of the arrays.

The second type of query is to answer the following question: if we list $2n$ integers, $p(a_1)$, $p(a_1)+p(a_2)$, ..., $p(a_1) + p(a_2) + \ldots + p(a_n)$ and $p(b_1)$, $p(b_1) + p(b_2)$, ..., $p(b_1) + p(b_2) + \ldots + p(b_n)$, and sort them into $c_1, c_2, \ldots, c_{2n}$, what would $p^{-1}(c_k)$ be? It is guaranteed that k is a **cute** number.

There are two interaction functions for you to call.

- add(x, y): returns $p^{-1}(p(x) + p(y))$.
- cmp(x, y): returns $p^{-1}(\min(p(x), p(y)))$.

Please beware that it is invalid to ask add(x, y) if $p(x) + p(y) \geq 2^{64}$.

To help you refrain from making invalid calls, it is guaranteed that, at any moment, the following condition holds: $\max(p(a_1) + p(a_2) + \ldots + p(a_n), p(b_1) + p(b_2) + \ldots + p(b_n)) < 2^{64}$.

Input

You begin the interaction by reading three integers: n, q, B ($1 \leq n \leq 1.6 \cdot 10^4$; $1 \leq q \leq 2 \cdot 10^4$; $1 \leq B \leq 16$).

Then, you should read two lines, the first containing the array $a_1, a_2, \ldots, a_n$ ($0 \leq a_i < 2^{64}$), and the second containing the array $b_1, b_2, \ldots, b_n$ ($0 \leq b_i < 2^{64}$).

After that, you should read the contents of the q queries.

For the next q lines, the first integer *type* indicates the type of query ($1 \leq type \leq 2$).

- If $type = 1$, three integers follow: t, pos, x ($1 \leq t \leq 2$; $1 \leq pos \leq n$; $0 \leq x < 2^{64}$).
  - If $t = 1$, set $a_{pos} = x$.
  - If $t = 2$, set $b_{pos} = x$.
- If $type = 2$, one integer k follows ($1 \leq k \leq \min(2^B - 1, 2 \cdot n)$). It is guaranteed that k is a **cute** number.

It is guaranteed that there is at least 1 and at most 5000 queries of type 1.

Interaction Protocol

After reading all the input, you can make at most $1.6 \cdot 10^6$ function calls.

For each call, you print a line of the form "t x y", where t is either "`A`" or "`C`", and x and y are integers ($0 \le x, y < 2^{64}$). Then flush the output, and read a line with the resulting integer z:

- If t is "`A`", you will get $z = p^{-1}(p(x) + p(y))$.
- If t is "`C`", you will get $z = p^{-1}(\min(p(x), p(y)))$.

If you make too many calls or make an invalid call, you will receive the `Wrong Answer` verdict.

After you have calculated the answers to all queries of type 2, print two lines. The first line should be "`!` m", where m is the number of type 2 queries. The second line should contain m integers $q_1, \ldots, q_m$: the answers to the type 2 queries respectively. Note that printing the answer does not count towards your total of $1.6 \cdot 10^6$ calls. After printing these two lines, your program should terminate.

Example

standard input	*standard output*
2 3 2 1 3 5 7 2 3 1 2 2 9 2 3 4 12 12 14 4	 A 1 3 A 5 7 C 4 12 A 5 9 C 4 14 ! 2 12 4