

Task Automatizacija

The lifelong ambition of successful entrepreneur Elena Mošus is to replace all human labor with artificial intelligence. To accelerate this process, she decided to get involved in Croatian legislation. Her initiative reached the president, who appointed her as the head of the newly formed Ministry of Automation of Logical Notions and Analytical Reasoning (abbreviated as MALNAR). Their first task is to automate the following game.

The game is played between two players. Each player receives a set of K distinct numbers between 1 and N . Players have access only to the numbers from their own set, and the goal of the game is to determine the size of the intersection of the two sets. Players cannot communicate directly — they may only communicate through a shared board by placing tokens on it. The rules of the game are as follows:

- The board consists of N empty fields where tokens can be placed.
- Players take turns placing tokens on any available field. Once a token is placed on a field, that field becomes occupied and cannot be used again.
- The first player places blue tokens, and the second player places red tokens.
- Both players have full visibility of the entire board at all times.
- On their turn, instead of placing a token, a player may choose to end the game by declaring the size of the intersection of the two sets. If there are no free fields remaining, the player must end the game.

During the game, players may communicate only through their moves on the board; however, before the game begins, they are allowed to agree on a strategy.

MALNAR has decided that the players in this game should be replaced by an automated artificial intelligence system using an infallible strategy, capable of making a move immediately after reading the current board state. Help MALNAR by designing a strategy for both players that guarantees that at least one player, at some point, ends the game and correctly declares the size of the intersection.

To enable the automated system to make moves quickly, the strategy must specify, for every possible board state, which move to make. This means that the system will not have access to the sequence of previous moves leading to the current state — it must decide solely based on the present board configuration.

Input

The first line contains a positive integer P ($P = 1$ or $P = 2$) indicating whether you are controlling the first or the second player.

The second line contains two positive integers N and K as described above.

The third line contains K distinct positive integers between 1 and N , representing the player's set.

The fourth line contains a positive integer T — the number of board states for which a move must be determined. In the test data, T will always equal the total number of possible board states, meaning your strategy must specify a move for every possible state. A board state is considered *possible* if it can arise during gameplay. However, in sample cases, T may be smaller.

Each of the next T lines describes one possible board state. A board state is given as a sequence of N characters “P”, “C”, or “.”, where “P” denotes a blue token, “C” denotes a red token, and “.” denotes an empty field.

Output

For each of the T given board states, output one line of the form “+ m ” or “! m ”, for some integer m .

An output of the form “+ m ” represents placing a token on the m -th field of the board. For the output to be valid, it must satisfy $1 \leq m \leq N$, and the chosen field must be unoccupied.



An output of the form “! m “ represents declaring that the size of the intersection of the two sets is m and ending the game. For the output to be valid, it must satisfy $0 \leq m \leq N$.

Scoring

Your solution will be evaluated in two stages. First, it will be tested with $P = 1$, and then with $P = 2$. The values of N and K will be identical in both tests. Assuming your program produces valid outputs for both stages, a simulator developed by the organizers will simulate the game according to your printed strategy. If the simulation concludes with the correct intersection size being declared, your solution will be considered correct. The total execution time will be the sum of the times for both stages.

In all subtasks, it holds that $2 \leq N \leq 16$ and $1 \leq K \leq N$.

Subtask	Points	Constraints
1	11	The sets will consist of K consecutive numbers.
2	7	N is even and all numbers in the sets are between 1 and $\frac{N}{2}$.
3	16	$N \leq 4$
4	13	$N = 14$ and $K = 2$
5	12	All numbers in the sets are between 1 and $N - 1$.
6	41	No additional constraints.

Sample Cases

input	input
1	2
4 2	4 2
2 3	1 3
3	2
....	P...
P.C.	P.CP
PCCP	output
output	+ 3
+ 1	+ 2
+ 4	
! 1	

Explanation of the Sample Cases:

This represents only one possible strategy. The corresponding course of the game is given below.

Board State	Move	Note
....	+ 1	The first player places a token on the first field.
P...	+ 3	The second player places a token on the third field.
P.C.	+ 4	The first player places a token on the fourth field.
P.CP	+ 2	The second player places a token on the second field.
PCCP	! 1	The first player ends the game and declares the intersection size to be 1. Correct!