

8190 Complete Naebbirac's sequence

Naebbirac is a young and easy-to-get-bored sailor. He likes sequences of integers and to come up with ways to classify them. Naebbirac says that a sequence is *complete* for a chosen integer K , if the sequence only contains integers between 1 and K , and each integer between 1 and K appears the same number of times.

Based on that, Naebbirac created a game to entertain himself and his peers, when the waters calm down and there's not much they can do to spend their time in the middle of the ocean.

First he chooses a positive integer K and then he uses chalk to draw on the deck a sequence S having N integers between 1 and K . After that he challenges one of his peers. The goal of the challenged peer is to turn the sequence S into a *complete* sequence by performing exactly one of the following three possible operations:

- '- x ': remove one occurrence of integer x from S ;
- '+ x ': add a new integer with value x in S ; or
- '- x + y ': replace one occurrence of integer x from S by an integer with value y .

Naebbirac is quite smart. He never writes a sequence that is already *complete* and often the written integers don't follow a pattern, making it quite hard to find an operation that solves the puzzle. One of your friends, that usually sails with Naebbirac, is tired of always losing the game. Are you able to help your friend and create a computer program that can find a solution to Naebbirac's game before they go on their next trip?

Input

The input file contains several test cases, each of them as described below.

The first line contains two integers K ($3 \leq K \leq 1000$) and N ($1 \leq N \leq 10^4$), indicating respectively the integer that Naebbirac chooses at the beginning of the game, and the length of the sequence written on the deck. The second line contains N integers S_1, S_2, \dots, S_N ($1 \leq S_i \leq K$ for $i = 1, 2, \dots, N$) representing the written sequence; you can safely assume that the sequence is not *complete*.

Output

For each test case, output a single line with the description of the operation that allows your friend to win the game or an '*' (asterisk) if there is no way to win. The description of the operation must follow the format shown on the statement, i.e. '- x ', '+ x ' or '- x + y '.

Sample Input

```
3 5
1 3 2 3 1
3 7
1 2 3 3 3 2 1
3 6
3 1 2 1 3 1
3 6
2 3 2 2 2 1
```

Sample Output

```
+2
-3
-1 +2
*
```