THE 13TH LATVIAN OLYMPIAD IN INFORMATICS
3RD STAGE PROBLEMS
Day 1 (march 27, 2000)

1. "Pack"

There is a pack of cards on the table, containing *n* cards placed one on another. Each card has a positive integer written on one side and nothing on the other. On the top card the number 1 is written, on the second from top number 2, etc., and on the bottom card the number *n* is written. In the beginning all the cards in the pack are placed so that numbers are facing up. Archibald makes *m* turns. In the *i*-th turn he takes the upper k_i cards, holding them together turns them upside down and places back on top of the pack. Your task is to write a program that determines the place and state of a card (upside up or down) in the pack after *m* moves made by Archibald!

Input data

The first line of the text file `KAVA.DAT` contains two positive integers – n (the number of cards in the pack, n≤100000) and m (the number of turns done by Archibald, m≤1000) – which are separated using a space symbol. Each of the following m lines contains one positive integer k_i ($1 \le k_i \le n$) – the number of cards used in each turn.

The first line of the text file `KAVA.TST` contains one positive integer s (s≤10000) – the number of cards, the final position and state of which must be determined. Each of the following s lines contains one positive integer – the number written on the card, the final position and state of which must be determined.

Output data

The text file `KAVA.REZ` must contain excatly s lines. Each line must contain one integer. If in the end the p-th card has the number that the i+1st line of the file `KAVA.TST` contains, the i-th line of the output file must contain the number

- **+p** if the card has its number written on the side facing up or
- **-p** if the card has its number written on the side facing down.

Example

Input data		Output data
File `KAVA.DAT`	File `KAVA.TST`	`File KAVA.REZ`
8 3	5	-5
1	4	+4
8	8	+8
4	1	+1
	5	-7
	2	

Note: The order of cards in the pack was changing thus (↑ shows that the card is placed with the number up, while ↓ – down):

```
1↑      1↓      8↓      5↑
2↑      2↑      7↓      6↑
3↑      3↑      6↓      7↑
4↑  ⇒   4↑  ⇒   5↓  ⇒   8↑
5↑      5↑      4↓      4↓
6↑      6↑      3↓      3↓
7↑      7↑      2↓      2↓
8↑      8↑      1↑      1↑
```

2. "Puzzle of numbers"

When a part of digits in a correct equation of sum of two numbers is replaced with stars (`*`) , a *puzzle of numbers* is formed. Thus, for example, if the starting equation was the following:

```
  9334
   789
 ------
 10123    (9334 + 789 = 10123),
```

then puzzles corresponding to it are:

```
   *3*4          9**4          ****
    78*           **9           ***
  ------       -------       -------
   10123        *****         *****    ,
```

as well as many more.

Your task is to write a program that would find the starting equation for a given puzzle. If there is more than one starting equation possible, your task is to find one of these.

It is known that all the puzzles given will have at least one solution and in the starting equation the first digit of any numbers is not equal to 0.

Input data

The text file `REBUSS.DAT` contains three lines. Each line contains a string containing digits and * signs. The length of any of the strings does not exceed 20 symbols. The first and the second line of the file contain the numbers to be added and the third line the sum.

Output data

The text file `REBUSS.REZ` must contain exactly three lines, corresponding to the starting equation. Each of these must contain one positive integer. The first and the second lines of the file must contain the numbers to be added and the third line the sum.

Example

Input data (file `REBUSS.DAT`)	Output data (file `REBUSS.REZ`)
`*3*4`	`9334`
`78*`	`789`
`10123`	`10123`