An Animal Contest 4 P1 - Dasher's Digits

Time limit: 2.0s **Memory limit:** 256M

Dasher the reindeer has a string X with N characters, where M characters are `0`'s and the rest are uppercase Latin letters. Each `0` is assigned a "cheer value" a_i. The cheer values are $a_1, a_2, \ldots, a_M$ where a_1 corresponds to the cheer value for the first `0` in X, a_2 the cheer value for the second `0`, and so on.

Dasher wants to get rid of all the `0`'s in his string, so he performs the following algorithm while there are still `0`'s present:

- If the frontmost character is `0`, subtract that `0`'s cheer value by 1. If this cheer value becomes 0, remove this `0`. Otherwise, move it, along with its cheer value, to the back of the string.
- Otherwise, take the frontmost character and move it to the back of the string.

For strings that are extensive in length, performing the algorithm manually is a tedious process. Dasher requires your assistance to output the string after all `0`'s have been removed through the usage of the mentioned algorithm.

Constraints

$2 \leq N \leq 10^6$

$1 \leq M < N$

$1 \leq a_i \leq 10^9$

X contains at least one non-`0` character.

Subtask 1 [20%]

$M = 1$

Subtask 2 [30%]

$M \leq 2$

Subtask 3 [50%]

No additional constraints.

Input Specification

The first line will contain two space-separated integers N and M.

The second line will contain X, which has at least one non-`0` character, and may contain leading `0`'s.

The next line will contain M space-separated integers a_i.

Output Specification

Output the string X after all `0`'s are removed with the algorithm described.

Sample Input

```
7 1
US0AMOG
2
```

Sample Output

```
AMOGUS
```

Explanation

The following shows the results of performing the algorithm manually to obtain our answer:

`US0AMOG` → `S0AMOGU` → `0AMOGUS` → `AMOGUS0` → `MOGUS0A` → `OGUS0AM` → `GUS0AMO` → `US0AMOG` → `S0AMOGU` → `0AMOGUS` → `AMOGUS`