

CST2016 1-2 Zuma

描述

祖玛是一款曾经风靡全球的游戏，其玩法是：在一条轨道上初始排列着n个彩色珠子，其中任意三个相邻的珠子不会完全同色。此后，你可以发射珠子到轨道上并加入原有序列中。一旦有三个或更多同色的珠子变成相邻，它们就会立即消失。这类消除现象可能会连锁式发生，其间你将暂时不能发射珠子。



开发商最近准备为玩家写一个游戏过程的回放工具。他们已经在游戏内完成了过程记录的功能，而结算功能的实现则委托你来完成。

游戏过程的记录中，首先是轨道上初始的珠子序列，然后是玩家接下来所做的一系列操作。你的任务是，计算经过一系列操作之后的珠子序列。

输入

第1行是一个由大写字母'A'~'Z'组成的字符串，表示轨道上初始的珠子序列，不同的字母表示不同的颜色。

第2行是一个数字m，表示整个回放过程共有m次操作。

接下来的m行依次对应于各次操作。每次操作由一个数字k和一个大写字母Σ描述，以空格分隔。其中，Σ为新珠子的颜色。若插入前共有b颗珠子，则 $k \in [0, b]$ 表示新珠子嵌入之后（尚未发生消除之前）在轨道上的位序。

输出

输出共1行，给出所有操作（及可能随即发生的消除现象）之后轨道上的珠子序列。

如果最终轨道上没有珠子，则仅输出一个空行（即仅输出一个换行符）。

输入样例

```
ACCBAA  
5  
1 B  
0 A  
2 B  
4 C  
0 A
```

*此样例是第1个测试点

输出样例

```
A
```

样例解释

各次操作后，轨道上的珠子序列分别是

```
ABCCBA  
AABCCBA  
AABBCCBA  
—  
A
```

数据范围

$0 \leq n \leq 1,000,000$

$0 \leq m \leq 500,000$

资源限制

时间限制：1 sec

空间限制：256 MB

提示

本题字母、数字混合输入，请使用足够鲁棒的读入方式。

- 最明显的思路是使用某种数据结构维护当前序列，对于各个操作进行相应的模拟操作。
- 参考习题 [2-23] a)，把轨道划分成多个部分，使珠子的定位、插入、消除都能达到较高的效率。