

C. 簡單 V 硬

Description

V 硬是世界上最好用的文字編輯器，不只能夠編輯文字，還可以自己新增許多快捷建，讓寫 Code 變得簡簡單單輕輕鬆鬆。醜緒在高中的時候，最喜歡用 V 硬來寫程式了。在高二那年獲得推薦資格後，閒閒沒事做，心血來潮想要實做看看簡單版本的 V 硬，他希望他的簡單 V 硬能夠支援底下的功能。

一開始簡單 V 硬會打開一個文字檔，這個文字檔總共有 N 行，第 i 行是一個字串，以 S_i 表示。在 V 硬裡面有一個剪貼簿 X 可以紀錄你上次剪下的東西。然後醜緒 Q 次操作，每次都是底下兩種之中的其中一種

1. `cut i l r` 把 S_i 的第 l 個字元到第 r 個字元剪下來放進剪貼簿。也就是說在做完這個操作後，新的 S'_i 會變成

$$S_{i,0}S_{i,1}\dots S_{i,l-1}S_{i,r+1}\dots S_{i,|S_i|-1}$$

且剪貼簿 X 會變成

$$S_{i,l}S_{i,l+1}\dots S_{i,r}$$

舉個簡單的例子，如果原本 S_2 為 abcdef，那經過 `cut 2 2 4` 之後，新的 S'_2 會變成 abf，而剪貼簿 X 會變成 cde

註：保證第一個操作一定是 `cut` 操作。

2. `paste i p` 把剪貼簿 X 裡面的東西插入到 S_i 的第 $p-1$ 個字元和第 p 個字元之間。也就是說在做完這個操作後，新的 S'_i 會變成

$$S_{i,0}S_{i,1}\dots S_{i,p-1}XS_{i,p}S_{i,p+1}\dots S_{i,|S_i|-1}$$

特別的，如果 $p=0$ ， $S'_i=XS$ ，而如果 $p=|S_i|$ ， $S'_i=SX$

註：在經過 `paste` 操作後，剪貼簿的內容並不會改變。

現在醜緒想要知道，經過上面所有的操作後，文字檔最後會變成什麼樣子。

Input

輸入的第一行有一個正整數 N, Q 接下來 N 行，每行有一個字串 S_i 代表文字檔第 i 行的樣子（注意到 S_i 可能是空字串）

接下來 Q 行，每行代表一個操作，格式如下

1. `cut i l r`

2. `paste i p`

- $1 \leq N \leq 100$
- $0 \leq i < N$
- $0 \leq |S_i| \leq 100$
- $0 \leq Q \leq 10000$
- $0 \leq l \leq r < |S_i|$
- $0 \leq p \leq |S_i|$
- S_i 僅包含空白、大小寫英文字母、數字
- 經過任意操作後， $|S_i| \leq 200$

註：當 K 是一個字串的時候 $|K|$ 代表的是字串 K 的長度。

Output

請輸出 N 行，其中第 i 行代表 S_i 。

注意：請勿輸出額外的行尾空白。

Sample 1

Input	Output
3 5 Hi Hi This is Sprout You are very welcome to use vim cut 2 24 27 cut 2 24 26 paste 0 20 cut 0 14 19 paste 2 24	Hi Hi This is vim You are very welcome to Sprout

配分

在一個子任務的「測試資料範圍」的敘述中，如果存在沒有提到範圍的變數，則此變數的範圍為 Input 所描述的範圍。

子任務編號	子任務配分	測試資料範圍
0	0%	範例測試資料
1	10%	S_i 內不包含空白，且 $0 < S_i $ ，只有 cut 操作
2	10%	只有 cut 操作
3	20%	S_i 內不包含空白，且 $0 < S_i $
4	10%	$Q \leq 30$
5	50%	無額外限制