

표 편집

Python3

문제 설명

[본 문제는 정확성과 효율성 테스트 각각 점수가 있는 문제입니다.]

업무용 소프트웨어를 개발하는 니니즈웍스의 인턴인 양문도는 명령어 기반으로 표의 행을 선택, 삭제, 복구하는 프로그램을 작성하는 과제를 맡았습니다. 세부 요구 사항은 다음과 같습니다

행 번호	이름
0	무지
1	콘
2	어피치
3	제이지
4	프로도
5	네오
6	튜브
7	라이언

위 그림에서 파란색으로 칠해진 칸은 현재 선택된 행을 나타냅니다. 단, 한 번에 한 행만 선택할 수 있으며, 표의 범위(0행 ~ 마지막 행)를 벗어날 수 없습니다. 이때, 다음과 같은 명령어를 이용하여 표를 편집합니다.

- 선택
  - "U X": 현재 선택된 행에서 X칸 위에 있는 행을 선택합니다.
  - "D X": 현재 선택된 행에서 X칸 아래에 있는 행을 선택합니다.
- 삭제
  - "C": 현재 선택된 행을 삭제한 후, 바로 아래 행을 선택합니다. 단, 삭제된 행이 가장 마지막 행인 경우 바로 윗 행을 선택합니다.
  - "Z": 가장 최근에 삭제된 행을 원래대로 복구합니다. 단, 현재 선택된 행은 바뀌지 않습니다.

예를 들어 위 표에서 "D 2"를 수행할 경우 아래 그림의 왼쪽처럼 4행이 선택되며, "C"를 수행하면 선택된 행을 삭제하고, 바로 아래 행이었던 "네오"가 적힌 행을 선택합니다(4행이 삭제되면서 아래 있던 행들이 하나씩 밀려 올라오고, 수정된 표에서 다시 4행을 선택하는 것과 동일합니다).

행 번호	이름
0	무지
1	콘
2	어피치
3	제이지
4	프로도
5	네오
6	튜브
7	라이언

행 번호	이름
0	무지
1	콘
2	어피치
3	제이지
4	네오
5	튜브
6	라이언

다음으로 "U 3"을 수행한 다음 "C"를 수행한 후의 표 상태는 아래 그림과 같습니다.

행 번호	이름
0	무지
1	콘
2	어피치
3	제이지
4	네오
5	튜브
6	라이언

행 번호	이름
0	무지
1	어피치
2	제이지
3	네오
4	튜브
5	라이언

다음으로 "D 4"를 수행한 다음 "C"를 수행한 후의 표 상태는 아래 그림과 같습니다. 5행이 표의 마지막 행 이므로, 이 경우 바로 윗 행을 선택하는 점에 주의합니다.

행 번호	이름
0	무지
1	어피치
2	제이지
3	네오
4	튜브
5	라이언

행 번호	이름
0	무지
1	어피치
2	제이지
3	네오
4	튜브

다음으로 "U 2"를 수행하면 현재 선택된 행은 2행이 됩니다.

행 번호	이름
0	무지
1	어피치
2	제이지
3	네오
4	튜브

위 상태에서 "Z"를 수행할 경우 가장 최근에 제거된 "라이언"이 적힌 행이 원래대로 복구됩니다.

행 번호	이름
0	무지
1	어피치
2	제이지
3	네오
4	튜브
5	라이언

다시한번 "Z"를 수행하면 그 다음으로 최근에 제거된 "콘"이 적힌 행이 원래대로 복구됩니다. 이때, 현재 선택된 행은 바뀌지 않는 점에 주의하세요.

행 번호	이름
0	무지
1	콘
2	어피치
3	제이지
4	네오
5	튜브
6	라이언

이때, 최종 표의 상태와 처음 주어진 표의 상태를 비교하여 삭제되지 않은 행은 "O", 삭제된 행은 "X"로 표시하면 다음과 같습니다.

행 번호	이름	비교
0	무지	O
1	콘	O
2	어피치	O
3	제이지	O
4	프로도	X
5	네오	O
6	튜브	O
7	라이언	O

처음 표의 행 개수를 나타내는 정수  $n$ , 처음에 선택된 행의 위치를 나타내는 정수  $k$ , 수행한 명령어들이 담긴 문자열 배열 `cmd`가 매개변수로 주어질 때, 모든 명령어를 수행한 후 표의 상태와 처음 주어진 표의 상태를 비교하여 삭제되지 않은 행은 O, 삭제된 행은 X로 표시하여 문자열 형태로 return 하도록 solution 함수를 완성해주세요.

## 제한사항

- $5 \leq n \leq 1,000,000$
- $0 \leq k < n$
- $1 \leq \text{cmd}$ 의 원소 개수  $\leq 200,000$ 
  - `cmd`의 각 원소는 "U X", "D X", "C", "Z" 중 하나입니다.
  - X는 1 이상 300,000 이하인 자연수이며 0으로 시작하지 않습니다.
  - X가 나타내는 자연수에 ',' 는 주어지지 않습니다. 예를 들어 123,456의 경우 123456으로 주어집니다.
  - `cmd`에 등장하는 모든 X들의 값을 합친 결과가 1,000,000 이하인 경우만 입력으로 주어집니다.
  - 표의 모든 행을 제거하여, 행이 하나도 남지 않는 경우는 입력으로 주어지지 않습니다.
  - 본문에서 각 행이 제거되고 복구되는 과정을 보다 자연스럽게 보이기 위해 "이름" 열을 사용하였으나, "이름" 열의 내용이 실제 문제를 푸는 과정에 필요하지는 않습니다. "이름" 열에는 서로 다른 이름들이 중복없이 채워져 있다고 가정하고 문제를 해결해 주세요.
- 표의 범위를 벗어나는 이동은 입력으로 주어지지 않습니다.
- 원래대로 복구할 행이 없을 때(즉, 삭제된 행이 없을 때) "Z"가 명령어로 주어지는 경우는 없습니다.
- 정답은 표의 0행부터  $n - 1$ 행까지에 해당되는 O, X를 순서대로 이어붙인 문자열 형태로 return 해주세요.

## 정확성 테스트 케이스 제한 사항

- $5 \leq n \leq 1,000$
- $1 \leq \text{cmd}$ 의 원소 개수  $\leq 1,000$

## 효율성 테스트 케이스 제한 사항

- 주어진 조건 외 추가 제한사항 없습니다.

## 입출력 예

n	k	cmd	result

8	2	["D 2", "C", "U 3", "C", "D 4", "C", "U 2", "Z", "Z"]	"0000X000"
8	2	["D 2", "C", "U 3", "C", "D 4", "C", "U 2", "Z", "Z", "U 1", "C"]	"00X0X000"

입출력 예 설명

## 입출력 예 #1

문제의 예시와 같습니다.

## 입출력 예 #2

다음은 9번째 명령어까지 수행한 후의 표 상태이며, 이는 입출력 예 #1과 같습니다.

행 번호	이름
0	무지
1	콘
2	어피치
3	제이지
4	네오
5	튜브
6	라이언

강조는 2번.

1 0 2  
2 1 3 → 2를 제거하면  
3 2 4      2의 pc의 front를 2의 front로 변경  
4 3 5      2의 front의 pc를 2의 pc로 변경.  
5 4 6

1 0 3 ← 2를 넣으면  
3 1 4      2의 자리를 채우면.  
4 3 5      2의 pc front를 채우면 → 2 1 3  
5 4 6      2의 pc의 front를 2로 바꾸면.  
2의 front의 pc를 2로 바꾸면.

- ① 배열로 표였는데 강제로 한영. 효율성 35점.
- ② 찾아왔다. linked list로 풀어야 한다는 것을 알았다.
- ③. 문제 풀려면 시간복잡도를 계산해 봐야 한다는 것을 느꼈다.

10번째 명령어 "U 1"을 수행하면 "어피치"가 적힌 2행이 선택되며, 마지막 명령어 "C"를 수행하면 선택된 행을 삭제하고, 바로 아래 행이었던 "제이지"가 적힌 행을 선택합니다.

행 번호	이름
0	무지
1	콘
2	어피치
3	제이지
4	네오
5	튜브
6	라이언

행 번호	이름
0	무지
1	콘
2	제이지
3	네오
4	튜브
5	라이언

따라서 처음 주어진 표의 상태와 최종 표의 상태를 비교하면 다음과 같습니다.

행 번호	이름	비교
0	무지	O
1	큰	O
2	어피치	X
3	제이지	O
4	프로도	X
5	네오	O
6	튜브	O
7	라이언	O

---

#### 제한시간 안내

- 정확성 테스트 : 10초
- 효율성 테스트 : 언어별로 작성된 정답 코드의 실행 시간의 적정 배수

---

#### solution.py

```
1 def solution(n, k, cmd):
2     answer = ''
3     return answer
```

---

#### 실행 결과

실행 결과가 여기에 표시됩니다.

---

초기화코드 실행제출 후 채점하기