**로드러너 1 경비로봇 인공지능 제작**

로드러너 1은 고전게임 로드러너를 넥슨에서 리메이크한 게임이다. 플레이어는 러너를 조종해서 경비로봇들을 따돌리고 맵에 뿌려진 골드를 모두 획득해 탈출하려 한다.



러너에게 가장 빨리 갈 수 있는 경로로 움직이는 경비로봇 AI를 만들어보자.  
현재 게임 맵의 상태와 러너의 위치, 경비로봇의 위치가 주어졌을 때 경비로봇이 어떤 순서로 행동을 취해야 러너에게 가장 빨리 갈 수 있는지 결정하면 된다. 주어진 경비로봇의 처음 위치는 항상 블록을 밟고 서 있고, 러너는 움직이지 않는다고 가정한다.

경비로봇의 이동 규칙은 다음과 같다.

* 게임 맵은 빈 공간, 사다리, 블록으로 이뤄져 있다.
* 블록으로는 걸어들어갈 수 없다.
* 블록을 밟고 서있다면 왼쪽 혹은 오른쪽으로 이동 가능하다.
* 사다리에 매달린 상태에서는 위 혹은 아래로도 이동이 가능하다.
* 사다리는 밟고 올라설 수 있으며, 이 경우 사다리를 타고 아래로 내려갈 수 있다.
* 지금 있는 자리가 사다리에 매달리지 않고 아래가 빈 공간이라면 아래로 떨어진다.
* 왼쪽 혹은 오른쪽으로 한 칸 이동하는 데 H의 시간이 걸린다. (사다리를 타고 있는 중에도 왼쪽 또는 오른쪽으로 이동이 가능하고 이때도 H의 시간이 걸린다.)
* 사다리를 타고 위 혹은 아래로 한 칸 이동하는 데 V의 시간이 걸린다.
* 한 칸 아래로 떨어지는 데 F의 시간이 걸린다.

  
[사다리는 밟고 올라설 수 있다]

  
[사다리의 중간에서 옆으로 이동해서 낙하해도 된다]

**입력 형식**

첫 줄에는 맵의 가로 크기 N과 세로 크기 M이 공백으로 구분되어 주어진다. N과 M은 3,000 보다 크지 않은 자연수이다.  
둘째 줄에 H, V, F가 공백으로 구분되어 주어진다. H, V, F은 109보다 크지 않은 자연수이다.  
이후 M개의 줄에 맵 정보가 맨 위부터 아래로 차례로 주어진다.  
맵 정보는 빈 공간은 ‘.’, 블록은 ‘B’, 사다리는 ‘H’, 러너의 위치는 ‘R’, 경비로봇의 위치는 ‘G’로 주어진다.

**출력 형식**

첫 줄에 경비로봇이 이동할 방향을 L(왼쪽), R(오른쪽), U(위쪽), D(아래쪽)으로 공백없이 출력한다.

  
낙하하는 경우도 출력해야 한다. 위 상황의 경우 출력은 R, D로 출력한다.  
사다리에서 오른쪽으로 이탈하는 과정이 R, 낙하가 D이다. 만약 여러 칸을 낙하하면 낙하하는 칸수만큼 D를 출력한다.

최단 시간에 러너에게 도달 가능한 경로가 여럿이라면 그 중 아무거나 하나만 출력하면 된다.  
러너에게 도달 불가능할 경우 X를 출력한다.

**입력 예제 1**

1. 5 5
2. 1 1 1
3. BBBBB
4. B.R.B
5. B.BHB
6. BG.HB
7. BBBBB

**출력 예제 1**

1. RRUUL

**기숙사 방 배정**

넥슨 게임 개발학교는 기숙사를 운영하고있다. 기숙사에는 N명의 학생들이 있다. 학생들은 1번부터 N번까지 번호가 붙어 있다고 한다. 이들이 잠을 자는 기숙사는 N개의 방으로 구성되어 한 명이 방 하나를 쓰도록 되어 있다. 기숙사의 전체 모양은 2진트리 모양이다. 즉, 루트인 방이 하나 있고, 왼쪽과 오른쪽 **자식**인 방이 각각 존재할 수 있다. 자식인 방이 존재하는 경우 복도로 직접 연결되어 있다. 자식인 방 다음에도 동일한 방식으로 자식인 방들이 존재할 수 있다. 자식 관계의 반대를 **부모**라고 부른다.

어느 날 교장선생님은 학생들이 잠을 잘 방을 다음과 같은 “규칙1”에 따라 배정하도록 지시하였다.

규칙1: 루트의 왼쪽 자식과 그 연결된 방들의 수가 i개이면, 1번부터 i번 학생의 방을 왼쪽 자식과 그 연결된 방들에 배정하고, 루트에는 i+1번을 배정하며, 루트의 오른쪽 자식과 그 연결된 방들에 i+2번부터 N번까지의 학생들을 배치한다. 모든 방에 루트와 동일한 규칙을 재귀적으로 적용한다.

8개의 방을 가진 기숙사에 규칙1을 이용해서 방을 배정한 결과는 다음 그림과 같다. 각 방에 적힌 번호는 학생의 번호임을 알 수 있다.

전달 실수로 학생들은 다음과 같은 “규칙2”에 따라 방을 배정하고 말았다.

규칙2: 루트의 왼쪽 자식과 그 연결된 방들의 수가 i개이면, 1번부터 i번 학생의 방을 왼쪽 자식과 그 연결된 방들에 배정하고, 루트의 오른쪽 자식과 그 연결된 방들에 i+1번부터 N-1번까지의 학생들을 배치하며, 루트에는 N번을 배정한다. 모든 방에 루트와 동일한 규칙을 재귀적으로 적용한다.

규칙 2에 따라 방을 배정한 결과는 위와 같을 것이다. 교장 선생님은 연락이 잘못된 것을 알고 원래 규칙1대로 모두 방을 옮기도록 하였다. 하지만, 몇몇 학생들은 이미 잠이 들어 버려서 방을 옮길 수가 없었다. (위의 예에서는 3번 학생이 잠이 들었다.) 학생들은 할 수 없이, 잠든 학생들을 그대로 둔 상태로 나머지 학생들만 규칙1을 적용하여 방을 옮기는 것으로 실행했다. 잠든 학생들이 방을 옮기지 않은 상태에서 규칙1을 적용하는 것은 다음과 같이 실행했다고 한다.

규칙3: 잠들지 않은 학생들은 규칙1에 따라 방이 배정되는 학생의 번호 순서대로 작은 번호의 학생들부터 방을 배정받는다. 단, 잠든 학생이 있는 방이 있는 경우 그 방을 건너뛰고 같은 순서를 유지하여 방을 배정받는다.

이 규칙은 잠든 학생이 없는 경우 규칙1과 같은 결과를 만들어 낸다는 것을 알 수 있다. 하지만 위의 예와 같이 한명의 잠든 학생이 있는 경우 다음과 같은 결과가 됨을 알 수 있다.

기숙사의 방의 배치와, 잠든 학생들의 번호를 입력으로 받아서 규칙2로 배치된 상태에서 규칙3으로 배치된 상태로 이동하기 위해 각 학생이 지나야 하는 최소 복도의 개수의 합을 계산하는 프로그램을 작성하라.

**입력 형식**

첫째 줄에 방의 개수를 나타내는 자연수 N과 잠든 학생의 수를 나타내는 K가 주어진다. (1 ≤ N ≤ 100,000, 0 ≤ K ≤ N) 둘째 줄에 방들의 연결 상태가 다음의 규칙으로 주어진다. 우선 루트는 항상 존재하므로 루트가 존재하는지의 여부는 주어지지 않는다. 루트부터 시작하여 **존재하는 방들에 대해** 레벨 순서로, 왼쪽과 오른쪽 자식의 존재 여부가 각각, 순서대로 0 혹은 1로 주어진다. 1인 경우 자식이 존재한다는 것이다. 레벨 순서란 루트에 가까운 방들이 우선하고, 루트에서 거리가 같은 경우 왼쪽부터 정한 순서를 의미한다. 세번째 줄에 K개의 잠든 학생의 번호가 주어진다. 주어지는 학생의 번호는 모두 다르다. K=0인 경우 세번째 줄은 빈 줄로 주어진다. 입력 방식에 대해서, 아래의 입력 예의 첫번째 경우가 그림의 예제에 해당하는 것이므로 주의 깊게 살펴보라.

**출력 형식**

첫 줄에 각 학생이 지나야 하는 최소 복도의 개수의 합을 출력한다.

**입력 예제 1**

1. 8 1
2. 1 1 0 0 1 1 1 0 1 1 0 0 0 0 0 0
3. 3

**출력 예제 1**

1. 14

**입력 예제 2**

1. 2 1
2. 0 1 0 0
3. 2

**출력 예제 2**

1. 0