문제 : 최소 비용의 경로의 찾아라.

문제 설명 :

$n \times m$ 격자가 있을 때 각 칸의 인덱스는 (0,0)부터 시작하여 (n-1, m-1)까지 주어진다. 예를 들어, 4×3 격자의 인덱스는 아래 그림과 같다.

0,0	0,1	0,2
1,0	1,1	1,2
2,0	2,1	2,2
3,0	3,1	3,2

4	1	-1
9	5	6
-1	1	8
7	3	2

우리는 (0,0) 칸에서 시작하여 (n-1, m-1) 칸으로 이동하고자 한다. 이동은 반드시 우측 또는 아래측으로만 허용된다. 현재의 격자 위치가 (i, j) 라면, 우측으로 이동한다는 말은 (i, j+1)으로 이동하는 것을 의미하고, 아래측으로 이동한다는 말은 (i+1, j)로 이동하는 것을 말한다.

격자의 각 칸에는 정수가 할당되어 있다. 이 값은 여러분이 격자를 이동하면서 각 칸에서 지불해야 하는 비용이라고 생각하면 된다. **정수값이 -1인 칸은 지나갈 수 없는 칸을 의미한다**. 예를 들어, 좌측 그림에서처럼 각 칸에 정수 값이 주어질 때, 값이 4인 (0,0)에서 출발하여 9->5->6->8->2의 경로를 따라 이동이 가능하고, 이 때 총 지불해야 하는 비용은 4+9+5+6+8+2=34가 된다. 만약 4->9->5->1->3->2의 경로를 따라 이동하면 24의 비용을 지불해야 한다. 또 다른 가능한 경로는 4->1->5->1->3->2로써 이 때의 비용은 16이 된다. 우리는 이동 가능한 모든 경로 중 가장 비용이 적게 드는 경로를 찾고자 한다.

입력 :

입력화일의 이름은 gridpath.inp 이다. 입력 파일의 첫째 줄에는 격자의 크기를 나타내는 n과 m이 주어지고 이어서 n줄에 걸쳐 격자의 각 칸에 할당될 정수 값이 주어진다. 각 줄에는 m개의 정수값이 주어진다. 각 정수값은 공백 문자로 구분되며, 각 값은 최대 1000이다. n,m은 각각 200이하이다.

출력 :

출력화일의 이름은 gridpath.out이다. 가능한 경로 중 비용이 가장 적게 드는 경로를 찾아 그 때의 비용을 첫 줄에 보이고, 이어서 $n+m-1$줄에 걸쳐 경로의 정보를 나타내는 칸의 인덱스를 출력한다. 출발은 (0,0)이며, 도착지는 (n-1,m-1)이다. 만약 최저 비용의 경로가 2개 이상 있다면 우측으로 이동하는 경로를 우선으로 선택한다. 만약 최저 비용 경로를 찾을 수 없다면, 즉 경로가 막혀 있으면 “No path."라고 출력한다.

예제 :

입력 예	입력 예에 대한 출력
4 3 4 1 -1 9 5 6 -1 1 8 7 3 2	16 (0 0) (0 1) (1 1) (2 1) (3 1) (3 2)

입력 예	입력 예에 대한 출력
6 4 2 13 6 1 7 -1 1 1 5 100 5 1 1 -1 7 -1 13 6 3 9 30 5 9 4	50 (0 0) (0 1) (0 2) (1 2) (2 2) (3 2) (4 2) (4 3) (5 3)

입력 예	입력 예에 대한 출력
6 4 2 13 6 1 7 -1 1 1 5 100 5 1 1 -1 -1 -1 -1 6 3 9 30 5 9 4	No path.

제한조건: 프로그램은 gridpath.{c,cpp,java}로 한다.