

3-1

음료수 얼려 먹기

문제

$N \times M$ 크기의 얼음 틀이 있다. 구멍이 뚫려 있는 부분은 0, 칸막이가 존재하는 부분은 1로 표시된다.

구멍이 뚫려 있는 부분끼리 상, 하, 좌, 우로 붙어 있는 경우 서로 연결되어 있는 것으로 간주한다.

이때 얼음 틀의 모양이 주어졌을 때 생성되는 총 아이스크림의 개수를 구하는 프로그램을 작성하라.

다음의 4×5 얼음 틀 예시에서는 아이스크림이 총 3개가 생성된다

00110	0	0	1	1	0
00011	0	0	0	1	1
11111	1	1	1	1	1
00000	0	0	0	0	0

입력

- 첫 번째 줄에 얼음 틀의 세로 길이 N 과 가로 길이 M 이 주어진다. ($1 \leq N, M \leq 1,000$)
- 두 번째 줄부터 $N + 1$ 번째 줄까지 얼음 틀의 형태가 주어진다.
- 이때 구멍이 뚫려있는 부분은 0, 그렇지 않은 부분은 1이다.

출력

한 번에 만들 수 있는 아이스크림의 개수를 출력한다.

입력 예시 1

```
4 5
00110
00011
11111
00000
```

출력 예시 1

```
3
```

입력 예시 2

```
15 14
00000111100000
1111101111110
11011101101110
11011101100000
11011111111111
11011111111100
11000000011111
01111111111111
00000000011111
01111111111000
00011111111000
00000001111000
1111111110011
11100011111111
11100011111111
```

출력 예시2

```
8
```

3-2

미로 탈출

문제

$N \times M$ 크기의 직사각형 형태의 미로에 여러 마리의 괴물이 있어 이를 피해 탈출해야 한다. 현재 위치는 (1, 1)이고 미로의 출구는 (N,M)의 위치에 존재하며 한 번에 한 칸씩 이동할 수 있다. 괴물이 있는 부분은 0으로, 괴물이 없는 부분은 1로 표시되어 있다. 미로는 반드시 탈출할 수 있는 형태로 제시된다. 탈출하기 위해 움직여야 하는 최소 칸의 개수를 구하라. 칸을 셀 때는 시작 칸과 마지막 칸을 모두 포함해서 계산한다.

입력

- 첫째 줄에 두 정수 N, M ($4 \leq N, M \leq 200$)이 주어진다. 다음 N 개의 줄에는 각각 M 개의 정수(0 혹은 1)로 미로의 정보가 주어진다. 각각의 수들은 공백 없이 붙여서 입력으로 제시된다. 또한 시작 칸과 마지막 칸은 항상 1이다.

출력

첫째 줄에 최소 이동 칸의 개수를 출력한다.

입력 예시

```
5 6
101010
111111
000001
111111
111111
```

출력 예시

```
10
```

3-3

특정 거리의 도시 찾기

정답 문답

☆

2 Silver II

[너비 우선 탐색](#) [다익스트라](#) [그래프 이론](#) [그래프 탐색](#)난이도 제공: solved.ac — [난이도 투표하러 가기](#)

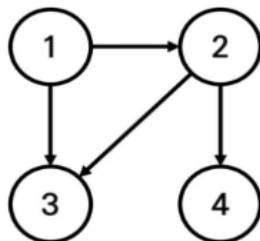
시간 제한	메모리 제한	제출	정답	맞은 사람	정답 비율
2 초	256 MB	6111	1730	1146	28.751%

문제

어떤 나라에는 1번부터 N 번까지의 도시와 M 개의 단방향 도로가 존재한다. 모든 도로의 거리는 1이다.

이 때 특정한 도시 X 로부터 출발하여 도달할 수 있는 모든 도시 중에서, 최단 거리가 정확히 K 인 모든 도시들의 번호를 출력하는 프로그램을 작성하시오. 또한 출발 도시 X 에서 출발 도시 X 로 가는 최단 거리는 항상 0이라고 가정한다.

예를 들어 $N=4$, $K=2$, $X=1$ 일 때 다음과 같이 그래프가 구성되어 있다고 가정하자.



이 때 1번 도시에서 출발하여 도달할 수 있는 도시 중에서, 최단 거리가 2인 도시는 4번 도시 뿐이다. 2번과 3번 도시의 경우, 최단 거리가 1이기 때문에 출력하지 않는다.

입력

첫째 줄에 도시의 개수 N , 도로의 개수 M , 거리 정보 K , 출발 도시의 번호 X 가 주어진다. ($2 \leq N \leq 300,000$, $1 \leq M \leq 1,000,000$, $1 \leq K \leq 300,000$, $1 \leq X \leq N$) 둘째 줄부터 M 개의 줄에 걸쳐서 두 개의 자연수 A , B 가 공백을 기준으로 구분되어 주어진다. 이는 A 번 도시에서 B 번 도시로 이동하는 단방향 도로가 존재한다는 의미다. ($1 \leq A, B \leq N$) 단, A 와 B 는 서로 다른 자연수이다.

출력

X 로부터 출발하여 도달할 수 있는 도시 중에서, 최단 거리가 K 인 모든 도시의 번호를 한 줄에 하나씩 오름차순으로 출력한다.

이 때 도달할 수 있는 도시 중에서, 최단 거리가 K 인 도시가 하나도 존재하지 않으면 -1을 출력한다.

예제 입력 1 복사

```
4 4 2 1
1 2
1 3
2 3
2 4
```

예제 출력 1 복사

```
4
```

예제 입력 2 복사

```
4 3 2 1
1 2
1 3
1 4
```

예제 출력 2 복사

```
-1
```

예제 입력 3 복사

```
4 4 1 1
1 2
1 3
2 3
2 4
```

예제 출력 3 복사

```
2
3
```

3-4

연구소

시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
2 초	512 MB	83980	48276	26851	54.798%

문제

인체에 치명적인 바이러스를 연구하던 연구소에서 바이러스가 유출되었다. 다행히 바이러스는 아직 퍼지지 않았고, 바이러스의 확산을 막기 위해서 연구소에 벽을 세우려고 한다.

연구소는 크기가 $N \times M$ 인 직사각형으로 나타낼 수 있으며, 직사각형은 1×1 크기의 정사각형으로 나누어져 있다. 연구소는 빈 칸, 벽으로 이루어져 있으며, 벽은 칸 하나를 가득 차지한다.

일부 칸은 바이러스가 존재하며, 이 바이러스는 상하좌우로 인접한 빈 칸으로 모두 퍼져나갈 수 있다. 새로 세울 수 있는 벽의 개수는 3개이며, 꼭 3개를 세워야 한다.

예를 들어, 아래와 같이 연구소가 생긴 경우를 살펴보자.

```
2 0 0 0 1 1 0
0 0 1 0 1 2 0
0 1 1 0 1 0 0
0 1 0 0 0 0 0
0 0 0 0 0 1 1
0 1 0 0 0 0 0
0 1 0 0 0 0 0
```

이때, 0은 빈 칸, 1은 벽, 2는 바이러스가 있는 곳이다. 아무런 벽을 세우지 않는다면, 바이러스는 모든 빈 칸으로 퍼져나갈 수 있다.

2행 1열, 1행 2열, 4행 6열에 벽을 세운다면 지도의 모양은 아래와 같아지게 된다.

```
2 1 0 0 1 1 0
1 0 1 0 1 2 0
0 1 1 0 1 0 0
0 1 0 0 0 1 0
0 0 0 0 0 1 1
0 1 0 0 0 0 0
0 1 0 0 0 0 0
```

바이러스가 퍼진 뒤의 모습은 아래와 같아진다.

```
2 1 0 0 1 1 2
1 0 1 0 1 2 2
0 1 1 0 1 2 2
0 1 0 0 0 1 2
0 0 0 0 0 1 1
0 1 0 0 0 0 0
0 1 0 0 0 0 0
```

벽을 3개 세운 뒤, 바이러스가 퍼질 수 없는 곳을 안전 영역이라고 한다. 위의 지도에서 안전 영역의 크기는 27이다.

연구소의 지도가 주어졌을 때 얻을 수 있는 안전 영역 크기의 최댓값을 구하는 프로그램을 작성하시오.

입력

첫째 줄에 지도의 세로 크기 N 과 가로 크기 M 이 주어진다. ($3 \leq N, M \leq 8$)

둘째 줄부터 N 개의 줄에 지도의 모양이 주어진다. 0은 빈 칸, 1은 벽, 2는 바이러스가 있는 위치이다. 2의 개수는 2보다 크거나 같고, 10보다 작거나 같은 자연수이다.

빈 칸의 개수는 3개 이상이다.

출력

첫째 줄에 얻을 수 있는 안전 영역의 최대 크기를 출력한다.

예제 입력 1 복사

```
7 7
2 0 0 0 1 1 0
0 0 1 0 1 2 0
0 1 1 0 1 0 0
0 1 0 0 0 0 0
0 0 0 0 0 1 1
0 1 0 0 0 0 0
0 1 0 0 0 0 0
```

예제 출력 1 복사

```
27
```

예제 입력 2 복사

```
4 6
0 0 0 0 0 0
1 0 0 0 0 2
1 1 1 0 0 2
0 0 0 0 0 2
```

예제 출력 2 복사

```
9
```

예제 입력 3 복사

```
8 8
2 0 0 0 0 0 0 2
2 0 0 0 0 0 0 2
2 0 0 0 0 0 0 2
2 0 0 0 0 0 0 2
2 0 0 0 0 0 0 2
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
```

예제 출력 3 복사

```
3
```

3-5

문제

- $N \times N$ 크기의 시험관이 있다.
- 시험관은 1×1 크기의 칸으로 나누어지며, 특정한 위치에는 바이러스가 존재할 수 있다.
- 모든 바이러스는 1번부터 K 번까지의 바이러스 종류 중 하나에 속한다.
- 시험관에 존재하는 모든 바이러스는 1초마다 상, 하, 좌, 우의 방향으로 증식해 나간다.
- 단, 매 초마다 번호가 낮은 종류의 바이러스부터 먼저 증식한다.
- 또한 증식 과정에서 특정한 칸에 이미 어떠한 바이러스가 존재한다면, 그 곳에는 다른 바이러스가 들어갈 수 없다.
- 시험관의 크기와 바이러스의 위치 정보가 주어졌을 때, S 초가 지난 후에 (X,Y) 에 존재하는 바이러스의 종류를 출력하는 프로그램을 작성하시오.
- 만약 S 초가 지난 후에 해당 위치에 바이러스가 존재하지 않는다면, 0을 출력한다.
- 이 때 X 와 Y 는 각각 행과 열의 위치를 의미하며, 시험관의 가장 왼쪽 위에 해당하는 곳은 $(1,1)$ 에 해당한다.
- 예를 들어 다음과 같이 3×3 크기의 시험관이 있다고 하자.
- 서로 다른 1번, 2번, 3번 바이러스가 각각 $(1,1)$, $(1,3)$, $(3,1)$ 에 위치해 있다.
- 이 때 2초가 지난 뒤에 $(3,2)$ 에 존재하는 바이러스의 종류를 계산해보자.

1		2
3		

- 1초가 지난 후에 시험관의 상태는 다음과 같다.

1	1	2
1		2
3	3	

- 2초가 지난 후에 시험관의 상태는 다음과 같다.

1	1	2
1	1	2
3	3	2

- 결과적으로 2초가 지난 뒤에 $(3,2)$ 에 존재하는 바이러스의 종류는 3번 바이러스다.
- 따라서 3을 출력하면 정답이다.

입력 조건

- 첫째 줄에 자연수 N, K 가 공백을 기준으로 구분되어 주어진다. ($1 \leq N \leq 200, 1 \leq K \leq 1,000$)
- 둘째 줄부터 N 개의 줄에 걸쳐서 시험관의 정보가 주어진다.
- 각 행은 N 개의 원소로 구성되며, 해당 위치에 존재하는 바이러스의 번호가 공백을 기준으로 구분되어 주어진다.
- 단, 해당 위치에 바이러스가 존재하지 않는 경우 0이 주어진다.
- 또한 모든 바이러스의 번호는 K 이하의 자연수로만 주어진다.
- $N+2$ 번째 줄에는 S, X, Y 가 공백을 기준으로 구분되어 주어진다. ($0 \leq S \leq 10,000, 1 \leq X, Y \leq N$)

출력 조건

- S 초 뒤에 (X, Y) 에 존재하는 바이러스의 종류를 출력한다.
- 만약 S 초 뒤에 해당 위치에 바이러스가 존재하지 않는다면, 0을 출력한다.

입력 예시

```
3 3
1 0 2
0 0 0
3 0 0
2 3 2
```

```
3 3
1 0 2
0 0 0
3 0 0
1 2 2
```

출력 예시

```
3
```

```
0
```


3-6



코딩테스트 연습 > 2020 KAKAO BLIND RECRUITMENT > 괄호 변환

괄호 변환

문제 설명

카카오에 신입 개발자로 입사한 "콘"은 선배 개발자로부터 개발역량 강화를 위해 다른 개발자가 작성한 소스 코드를 분석하여 문제점을 발견하고 수정 하라는 업무 과제를 받았습니다. 소스를 컴파일하여 로그를 보니 대부분 소스 코드 내 작성된 괄호가 개수는 맞지만 짝이 맞지 않은 형태로 작성되어 오류가 나는 것을 알게 되었습니다.

수정해야 할 소스 파일이 너무 많아서 고민하던 "콘"은 소스 코드에 작성된 모든 괄호를 뽑아서 올바른 순서대로 배치된 괄호 문자열을 알려주는 프로그램과 다음과 같이 개발하려고 합니다.

용어의 정의

'(' 와 ')' 로만 이루어진 문자열이 있을 경우, '(' 의 개수와 ')' 의 개수가 같다면 이를 **균형잡힌 괄호 문자열** 이라고 부릅니다.

그리고 여기에 '(' 와 ')' 의 괄호의 짝도 모두 맞을 경우에는 이를 **올바른 괄호 문자열** 이라고 부릅니다.

예를 들어, "(()())(" 와 같은 문자열은 "균형잡힌 괄호 문자열" 이지만 "올바른 괄호 문자열"은 아닙니다.

반면에 "(()())()" 와 같은 문자열은 "균형잡힌 괄호 문자열" 이면서 동시에 "올바른 괄호 문자열" 입니다.

'(' 와 ')' 로만 이루어진 문자열 w가 "균형잡힌 괄호 문자열" 이라면 다음과 같은 과정을 통해 "올바른 괄호 문자열"로 변환할 수 있습니다.

1. 입력이 빈 문자열인 경우, 빈 문자열을 반환합니다.
2. 문자열 w를 두 "균형잡힌 괄호 문자열" u, v로 분리합니다. 단, u는 "균형잡힌 괄호 문자열"로 더 이상 분리할 수 없어야 하며, v는 빈 문자열이 됩니다.
3. 문자열 u가 "올바른 괄호 문자열" 이라면 문자열 v에 대해 1단계부터 다시 수행합니다.
 - 3-1. 수행한 결과 문자열을 u에 이어 붙인 후 반환합니다.
4. 문자열 u가 "올바른 괄호 문자열"이 아니라면 아래 과정을 수행합니다.
 - 4-1. 빈 문자열에 첫 번째 문자로 '('를 붙입니다.
 - 4-2. 문자열 v에 대해 1단계부터 재귀적으로 수행한 결과 문자열을 이어 붙입니다.
 - 4-3. ')'를 다시 붙입니다.
 - 4-4. u의 첫 번째와 마지막 문자를 제거하고, 나머지 문자열의 괄호 방향을 뒤집어서 뒤에 붙입니다.
 - 4-5. 생성된 문자열을 반환합니다.

"균형잡힌 괄호 문자열" p가 매개변수로 주어질 때, 주어진 알고리즘을 수행해 "올바른 괄호 문자열"로 변환한 결과를 return 하도록 solution 함수를 완성해 주세요.

매개변수 설명

- p는 '(' 와 ')' 로만 이루어진 문자열이며 길이는 2 이상 1,000 이하인 짝수입니다.
- 문자열 p를 이루는 '(' 와 ')' 의 개수는 항상 같습니다.
- 만약 p가 이미 "올바른 괄호 문자열"이라면 그대로 return 하면 됩니다.

입출력 예

p	result
"((()))()"	"((()))()"
")(" "	"()"
"()))((())"	"()((()))"

입출력 예에 대한 설명

입출력 예 #1

이미 "올바른 괄호 문자열" 입니다.

입출력 예 #2

- 두 문자열 u, v로 분리합니다.
 - u = ")(" "
 - v = ""
- u가 "올바른 괄호 문자열"이 아니므로 다음과 같이 새로운 문자열을 만듭니다.
 - v에 대해 1단계부터 재귀적으로 수행하면 빈 문자열이 반환됩니다.
 - u의 앞뒤 문자를 제거하고, 나머지 문자의 괄호 방향을 뒤집으면 "" 이 됩니다.
 - 따라서 생성되는 문자열은 "(" + "" + ")" + ""이며, 최종적으로 "()"로 변환됩니다.

입출력 예 #3

- 두 문자열 u, v로 분리합니다.
 - u = "()" "
 - v = "))))((" "
- 문자열 u가 "올바른 괄호 문자열"이므로 그대로 두고, v에 대해 재귀적으로 수행합니다.
- 다시 두 문자열 u, v로 분리합니다.
 - u = "))))(" "
 - v = "()" "
- u가 "올바른 괄호 문자열"이 아니므로 다음과 같이 새로운 문자열을 만듭니다.
 - v에 대해 1단계부터 재귀적으로 수행하면 "()" 이 반환됩니다.
 - u의 앞뒤 문자를 제거하고, 나머지 문자의 괄호 방향을 뒤집으면 "()" 이 됩니다.
 - 따라서 생성되는 문자열은 "(" + "()" + ")" + "()"이며, 최종적으로 "((()))()"를 반환합니다.
- 처음에 그대로 둔 문자열에 반환된 문자열을 이어 붙이면 "()" + "((()))()" = "()((()))()"가 됩니다.

3-7

연산자 끼워넣기

시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
2 초	512 MB	88064	40043	26785	47.786%

문제

N 개의 수로 이루어진 수열 A_1, A_2, \dots, A_N 이 주어진다. 또, 수와 수 사이에 끼워넣을 수 있는 $N-1$ 개의 연산자가 주어진다. 연산자는 덧셈(+), 뺄셈(-), 곱셈(\times), 나눗셈(\div)으로만 이루어져 있다.

우리는 수와 수 사이에 연산자를 하나씩 넣어서, 수식을 하나 만들 수 있다. 이때, 주어진 수의 순서를 바꾸면 안 된다.

예를 들어, 6개의 수로 이루어진 수열이 1, 2, 3, 4, 5, 6이고, 주어진 연산자가 덧셈(+) 2개, 뺄셈(-) 1개, 곱셈(\times) 1개, 나눗셈(\div) 1개인 경우에는 총 60가지의 식을 만들 수 있다. 예를 들어, 아래와 같은 식을 만들 수 있다.

- $1+2+3-4\times 5\div 6$
- $1\div 2+3+4-5\times 6$
- $1+2\div 3\times 4-5+6$
- $1\div 2\times 3-4+5+6$

식의 계산은 연산자 우선 순위를 무시하고 앞에서부터 진행해야 한다. 또, 나눗셈은 정수 나눗셈으로 몫만 취한다. 음수를 양수로 나눌 때는 C++14의 기준을 따른다. 즉, 양수로 바꾼 뒤 몫을 취하고, 그 몫을 음수로 바꾼 것과 같다. 이에 따라서, 위의 식 4개의 결과를 계산해보면 아래와 같다.

- $1+2+3-4\times 5\div 6 = 1$
- $1\div 2+3+4-5\times 6 = 12$
- $1+2\div 3\times 4-5+6 = 5$
- $1\div 2\times 3-4+5+6 = 7$

N 개의 수와 $N-1$ 개의 연산자가 주어졌을 때, 만들 수 있는 식의 결과가 최대인 것과 최소인 것을 구하는 프로그램을 작성하시오.

입력

첫째 줄에 수의 개수 N ($2 \leq N \leq 11$)가 주어진다. 둘째 줄에는 A_1, A_2, \dots, A_N 이 주어진다. ($1 \leq A_i \leq 100$) 셋째 줄에는 합이 $N-1$ 인 4개의 정수가 주어지는데, 차례대로 덧셈(+)의 개수, 뺄셈(-)의 개수, 곱셈(\times)의 개수, 나눗셈(\div)의 개수이다.

출력

첫째 줄에 만들 수 있는 식의 결과의 최댓값을, 둘째 줄에는 최솟값을 출력한다. 연산자를 어떻게 끼워넣어도 항상 -10억보다 크거나 같고, 10억보다 작거나 같은 결과가 나오는 입력만 주어진다. 또한, 앞에서부터 계산했을 때, 중간에 계산되는 식의 결과도 항상 -10억보다 크거나 같고, 10억보다 작거나 같다.

예제 입력 1 복사

```
2
5 6
0 0 1 0
```

예제 출력 1 복사

```
30
30
```

예제 입력 2 복사

```
3
3 4 5
1 0 1 0
```

예제 출력 2 복사

```
35
17
```

예제 입력 3 복사

```
6
1 2 3 4 5 6
2 1 1 1
```

예제 출력 3 복사

```
54
-24
```

힌트

세 번째 예제의 경우에 다음과 같은 식이 최댓값/최솟값이 나온다.

- 최댓값: $1-2+3+4+5\times 6$
- 최솟값: $1+2+3+4-5\times 6$

3-8

감시 피하기

시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
2 초	256 MB	10212	4494	2979	43.337%

문제

$N \times N$ 크기의 복도가 있다. 복도는 1×1 크기의 칸으로 나누어지며, 특정한 위치에는 선생님, 학생, 혹은 장애물이 위치할 수 있다. 현재 몇 명의 학생들은 수업시간에 몰래 복도로 빠져나왔는데, 복도로 빠져나온 학생들은 선생님의 감시에 들리지 않는 것이 목표이다.

각 선생님들은 자신의 위치에서 상, 하, 좌, 우 4가지 방향으로 감시를 진행한다. 단, 복도에 장애물이 위치한 경우, 선생님은 장애물 뒤편에 숨어 있는 학생들은 볼 수 없다. 또한 선생님은 상, 하, 좌, 우 4가지 방향에 대하여, 아무리 멀리 있더라도 장애물로 막히기 전까지의 학생들은 모두 볼 수 있다고 가정하자.

다음과 같이 3×3 크기의 복도의 정보가 주어진 상황을 확인해보자. 본 문제에서 위치 값을 나타낼 때는 (행, 열)의 형태로 표현한다. 선생님이 존재하는 칸은 T, 학생이 존재하는 칸은 S, 장애물이 존재하는 칸은 O로 표시하였다. 아래 그림과 같이 (3,1)의 위치에는 선생님이 존재하며 (1,1), (2,1), (3,3)의 위치에는 학생이 존재한다. 그리고 (1,2), (2,2), (3,2)의 위치에는 장애물이 존재한다.

S	O	
S	O	
T	O	S

이 때 (3,3)의 위치에 존재하는 학생은 장애물 뒤편에 숨어 있기 때문에 감시를 피할 수 있다. 하지만 (1,1)과 (2,1)의 위치에 존재하는 학생은 선생님에게 들리게 된다. 학생들은 복도의 빈 칸 중에서 장애물을 설치할 위치를 골라, 정확히 3개의 장애물을 설치해야 한다. 결과적으로 3개의 장애물을 설치하여 모든 학생들을 감시로부터 피하도록 할 수 있는지 계산하고자 한다. $N \times N$ 크기의 복도에서 학생 및 선생님의 위치 정보가 주어졌을 때, 장애물을 정확히 3개 설치하여 모든 학생들이 선생님들의 감시를 피하도록 할 수 있는지 출력하는 프로그램을 작성하시오.

예를 들어 $N=5$ 일 때, 다음과 같이 선생님 및 학생의 위치 정보가 주어졌다고 가정하자.

	S			T
T		S		
	T			
		T		

이 때 다음과 같이 3개의 장애물을 설치하면, 모든 학생들을 선생님의 감시로부터 피하도록 만들 수 있다.

	S		O	T
T	O	S		
		O		
	T			
		T		

입력

첫째 줄에 자연수 N 이 주어진다. ($3 \leq N \leq 6$) 둘째 줄에 N 개의 줄에 걸쳐서 복도의 정보가 주어진다. 각 행에서는 N 개의 원소가 공백을 기준으로 구분되어 주어진다. 해당 위치에 학생이 있다면 S, 선생님이 있다면 T, 아무것도 존재하지 않는다면 X가 주어진다.

단, 전체 선생님의 수는 5이하의 자연수, 전체 학생의 수는 30이하의 자연수이며 항상 빈 칸의 개수는 3개 이상으로 주어진다.

출력

첫째 줄에 정확히 3개의 장애물을 설치하여 모든 학생들을 감시로부터 피하도록 할 수 있는지의 여부를 출력한다. 모든 학생들을 감시로부터 피하도록 할 수 있다면 "YES", 그렇지 않다면 "NO"를 출력한다.

예제 입력 1 복사

```
5
X S X X T
T X S X X
X X X X X
X T X X X
X X T X X
```

예제 출력 1 복사

```
YES
```

예제 입력 2 복사

```
4
S S S T
X X X X
X X X X
T T T X
```

예제 출력 2 복사

```
NO
```

3-9

인구 이동

시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
2 초	512 MB	59013	23916	13962	37.382%

문제

$N \times N$ 크기의 땅이 있고, 땅은 1×1 개의 칸으로 나누어져 있다. 각각의 땅에는 나라가 하나씩 존재하며, r 행 c 열에 있는 나라에는 $A[r][c]$ 명이 살고 있다. 인접한 나라 사이에는 국경선이 존재한다. 모든 나라는 1×1 크기이기 때문에, 모든 국경선은 정사각형 형태이다.

오늘부터 인구 이동이 시작되는 날이다.

인구 이동은 하루 동안 다음과 같이 진행되고, 더 이상 아래 방법에 의해 인구 이동이 없을 때까지 지속된다.

- 국경선을 공유하는 두 나라의 인구 차이가 L 명 이상, R 명 이하라면, 두 나라가 공유하는 국경선을 오늘 하루 동안 연다.
- 위의 조건에 의해 열어야 하는 국경선이 모두 열렸다면, 인구 이동을 시작한다.
- 국경선이 열려있어 인접한 칸만을 이용해 이동할 수 있으면, 그 나라를 오늘 하루 동안은 연합이라고 한다.
- 연합을 이루고 있는 각 칸의 인구수는 (연합의 인구수) / (연합을 이루고 있는 칸의 개수)가 된다. 편의상 소수점은 버린다.
- 연합을 해체하고, 모든 국경선을 닫는다.

각 나라의 인구수가 주어졌을 때, 인구 이동이 며칠 동안 발생하는지 구하는 프로그램을 작성하시오.

입력

첫째 줄에 N, L, R 이 주어진다. ($1 \leq N \leq 50, 1 \leq L \leq R \leq 100$)

둘째 줄부터 N 개의 줄에 각 나라의 인구수가 주어진다. r 행 c 열에 주어지는 정수는 $A[r][c]$ 의 값이다. ($0 \leq A[r][c] \leq 100$)

인구 이동이 발생하는 일수가 2,000번 보다 작거나 같은 입력만 주어진다.

출력

인구 이동이 며칠 동안 발생하는지 첫째 줄에 출력한다.

예제 입력 1 복사

```
2 20 50
50 30
20 40
```

예제 출력 1 복사

```
1
```

초기 상태는 아래와 같다.

50	30
20	40

L = 20, R = 50 이기 때문에, 모든 나라 사이의 국경선이 열린다. (열린 국경선은 점선으로 표시)

50	30
20	40

연합은 하나 존재하고, 연합의 인구는 (50 + 30 + 20 + 40) 이다. 연합의 크기가 4이기 때문에, 각 칸의 인구수는 $140/4 = 35$ 명이 되어야 한다.

35	35
35	35

예제 입력 2 복사

```
2 40 50
50 30
20 40
```

경계를 공유하는 나라의 인구 차이가 모두 L보다 작아서 인구 이동이 발생하지 않는다.

예제 입력 3 복사

```
2 20 50
50 30
30 40
```

예제 출력 2 복사

```
0
```

예제 출력 3 복사

```
1
```


초기 상태는 아래와 같다.

50	30
30	40

L = 20, R = 50이기 때문에, 아래와 같이 국경선이 열린다.

50	30
30	40

인구 수는 합쳐져있는 연합의 인구수는 $(50+30+30) / 3 = 36$ (소수점 버림)이 되어야 한다.

36	36
36	40

예제 입력 4 복사

```
3 5 10
10 15 20
20 30 25
40 22 10
```

예제 출력 4 복사

```
2
```

예제 입력 5 복사

```
4 10 50
10 100 20 90
80 100 60 70
70 20 30 40
50 20 100 10
```

예제 출력 5 복사

```
3
```

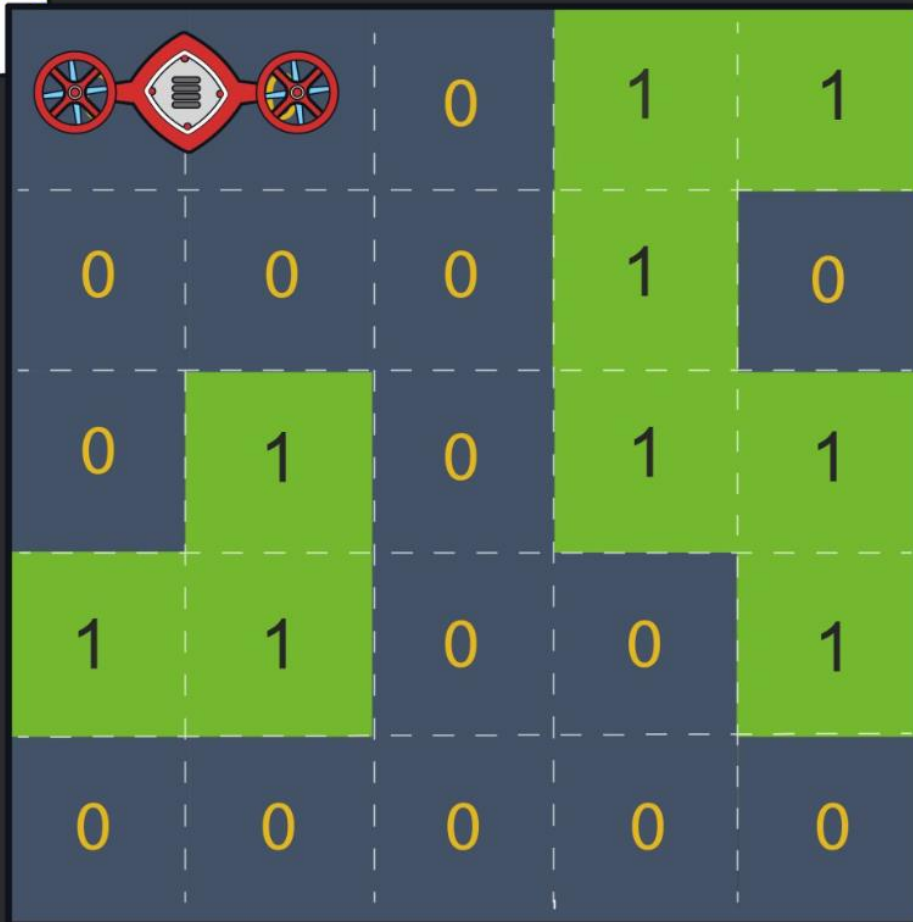
3-10

코딩테스트 연습 > 2020 KAKAO BLIND RECRUITMENT > 블록 이동하기

블록 이동하기

문제 설명

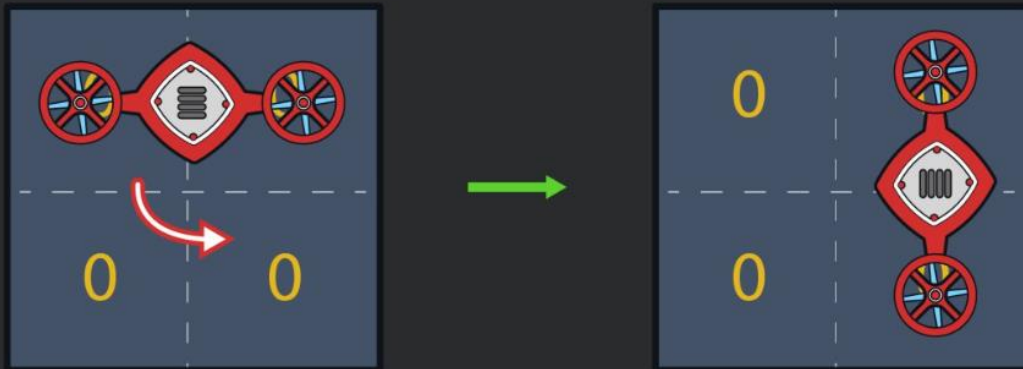
로봇개발자 "무지"는 한 달 앞으로 다가온 "카카오배 로봇경진대회"에 출품할 로봇을 준비하고 있습니다. 준비 중인 로봇은 2×1 크기의 로봇으로 "무지"는 "0"과 "1"로 이루어진 $N \times N$ 크기의 지도에서 2×1 크기인 로봇을 움직여 (N, N) 위치까지 이동 할 수 있도록 프로그래밍을 하려고 합니다. 로봇이 이동하는 지도는 가장 왼쪽, 상단의 좌표를 $(1, 1)$ 로 하며 지도 내에 표시된 숫자 "0"은 빈칸을 "1"은 벽을 나타냅니다. 로봇은 벽이 있는 칸 또는 지도 밖으로는 이동할 수 없습니다. 로봇은 처음에 아래 그림과 같이 좌표 $(1, 1)$ 위치에서 가로방향으로 놓여있는 상태로 시작하며, 앞뒤 구분 없이 움직일 수 있습니다.



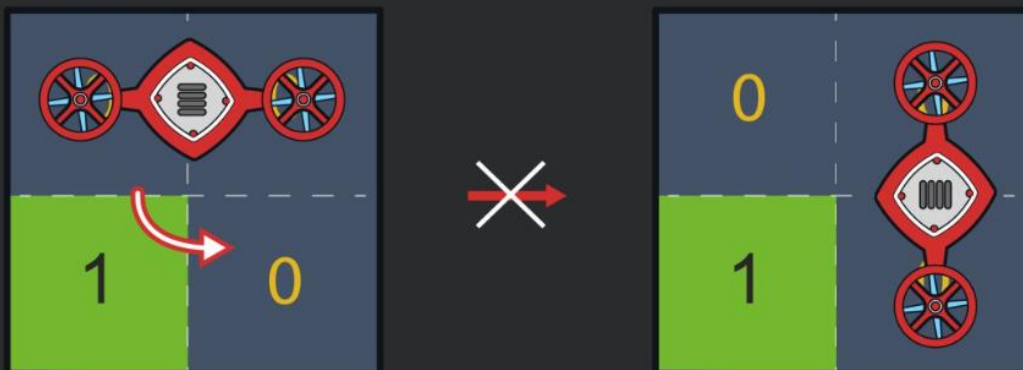
로봇이 움직일 때는 현재 놓여있는 상태를 유지하면서 이동합니다. 예를 들어, 위 그림에서 오른쪽으로 한 칸 이동한다면 (1, 2), (1, 3) 두 칸을 차지하게 되며, 아래로 이동한다면 (2, 1), (2, 2) 두 칸을 차지하게 됩니다. 로봇이 차지하는 두 칸 중 어느 한 칸이라도 (N, N) 위치에 도착하면 됩니다.

로봇은 다음과 같이 조건에 따라 회전이 가능합니다.

(A) ○



(B) ✕



위 그림과 같이 로봇은 90도씩 회전할 수 있습니다. 단, 로봇이 차지하는 두 칸 중, 어느 칸이든 축이 될 수 있지만, 회전하는 방향(축이 되는 칸으로부터 대각선 방향에 있는 칸)에는 벽이 없어야 합니다. 로봇이 한 칸 이동하거나 90도 회전하는 데는 걸리는 시간은 정확히 1초입니다.

"0"과 "1"로 이루어진 지도인 board가 주어질 때, 로봇이 (N, N) 위치까지 이동하는데 필요한 최소 시간을 return 하도록 solution 함수를 완성해주세요.

제한사항

- board의 한 변의 길이는 5 이상 100 이하입니다.
- board의 원소는 0 또는 1입니다.
- 로봇이 처음에 놓여 있는 칸 (1, 1), (1, 2)는 항상 0으로 주어집니다.
- 로봇이 항상 목적지에 도착할 수 있는 경우만 입력으로 주어집니다.

입출력 예

board	result
[[0, 0, 0, 1, 1],[0, 0, 0, 1, 0],[0, 1, 0, 1, 1],[1, 1, 0, 0, 1],[0, 0, 0, 0, 0]]	7

입출력 예에 대한 설명

문제에 주어진 예시와 같습니다.
로봇이 오른쪽으로 한 칸 이동 후, (1, 3) 칸을 축으로 반시계 방향으로 90도 회전합니다. 다시, 아래쪽으로 3칸 이동하면 로봇은 (4, 3), (5, 3) 두 칸을 차지하게 됩니다. 이제 (5, 3)을 축으로 시계 방향으로 90도 회전 후, 오른쪽으로 한 칸 이동하면 (N, N)에 도착합니다. 따라서 목적지에 도달하기까지 최소 7초가 걸립니다.