6680번 - 아깝게 놓친 COSS 장학금

시간 제한	메모리 제한	제출	정답	맞은 사람	정답 비율
1 초	1024 MB	0	0	0	0.000%

문제

COSS 사업단은 데이터 및 보안과 연관된 프로젝트를 진행한 학생들 중에서 가장 우수하게 프로젝트를 진행한 학생에게 장학금을 지급하고자 한다.

COSS 사업단에서 장학금 지급을 담당하는 재민이는 n명의 학생이 제출한 프로젝트를 평가한 후, 이들의 이름 name과 심사한 프로젝트 점수 score, 보안 위험도 risk, 프로젝트를 진행하는 데 사용된 비용 cost를 엑셀에 정리하였다.

COSS 사업단은 이렇게 정리한 데이터를 이용하여 가장 우수한 학생을 결정하는 **장학 점수**를 계산하고, 이 **장학 점수**가 가장 높은 학생에게 장학금을 지급하기로 했다.

이때, 각 학생의 **장학 점수**는 다음과 같이 계산된다.

$$\left\lfloor \frac{\text{score}^3}{\text{cost} \times (\text{risk} + 1)} \right\rfloor$$

그러나, 재민이는 **장학 점수**가 두 번째로 높은 학생도 열심히 노력했지만 장학금을 받지 못한 것이 아쉽다고 생각해 별도의 선물을 주려고 한다.

다시 엑셀을 열어서 선물을 줄 학생을 찾기에는 이미 지친 재민이를 도와서 **장학 점수**가 두 번째로 높은 학생의 이름을 찾아주자!

입력

첫 번째 줄에 학생의 수를 나타내는 정수 n이 주어진다. $(2 \le n \le 30)$

이후 n개의 줄에는 각 학생의 $\text{name}, \text{score}, \text{risk}, \text{cost}$ 정보가 공백으로 구분되어 주어진다.

여기서 name은 대문자 알파벳으로 이루어진 문자열이며, name의 길이는 2 이상 10 이하이다.

서로 다른 두 학생의 이름이 같은 경우는 주어지지 않는다.

score는 10 이상 100 이하의 정수, risk는 0 이상 9 이하의 정수, cost는 10 이상 200 이하의 정수이다.

출력

학생들을 다음과 같은 순서대로 나열할 때, 앞에서 두 번째로 오는 학생의 이름을 출력한다.

1. 장학 점수가 높은 학생이 먼저 오도록 내림차순으로 정렬한다.
2. 장학 점수가 동일한 학생들이 있는 경우, 프로젝트 진행 비용이 적은 학생이 먼저 오도록 오름차순으로 정렬한다.
3. 장학 점수와 프로젝트 진행 비용이 모두 동일한 학생들이 있는 경우, 이름이 사전 순으로 빠른 학생이 먼저 오도록 정렬한다.

예제 입력 1 복사

```
5
KIM 50 1 100
PARK 70 2 130
LEE 60 0 120
CHOI 50 1 90
JUNG 80 3 160
```

예제 출력 1 복사

```
PARK
```

예제 입력 2 복사

```
4
AA 30 2 20
BB 20 1 80
CC 10 1 10
DD 10 1 10
```

예제 출력 2 복사

```
CC
```

노트

① $\lfloor x \rfloor$는 x보다 작거나 같은 정수 중 가장 큰 값을 의미한다.

② "사전 순" 정렬 기준 : 문자열을 사전 순으로 정렬할 때는 다음 기준에 따라 순서를 결정한다.

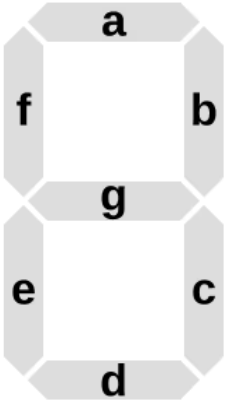
1. 문자열을 정렬할 때, 앞 글자부터 차례대로 비교하며 알파벳 순서 상 앞선 글자가 먼저 오는 순서가 우선이다.
2. 비교 중인 글자가 같을 경우, 그다음 글자를 계속 비교하여 순서를 결정한다.
3. 문자열의 길이가 다를 경우, 앞부분이 모두 같다면 짧은 문자열이 먼저 온다. (예: "ab" → "abc")

6686번 - 딸깍

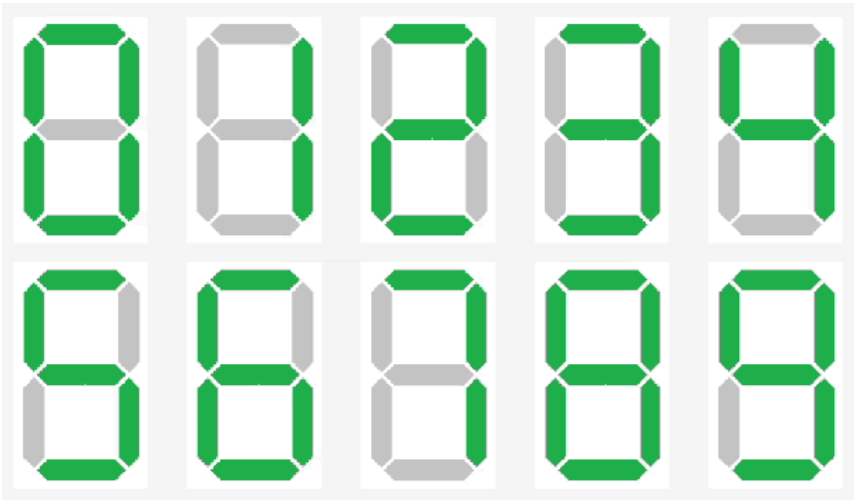
시간 제한	메모리 제한	제출	정답	맞은 사람	정답 비율
2 초	1024 MB	0	0	0	0.000%

문제

7-세그먼트 디스플레이는 아래 그림과 같이 7개의 세그먼트 a, b, c, d, e, f, g로 이루어져 있다.



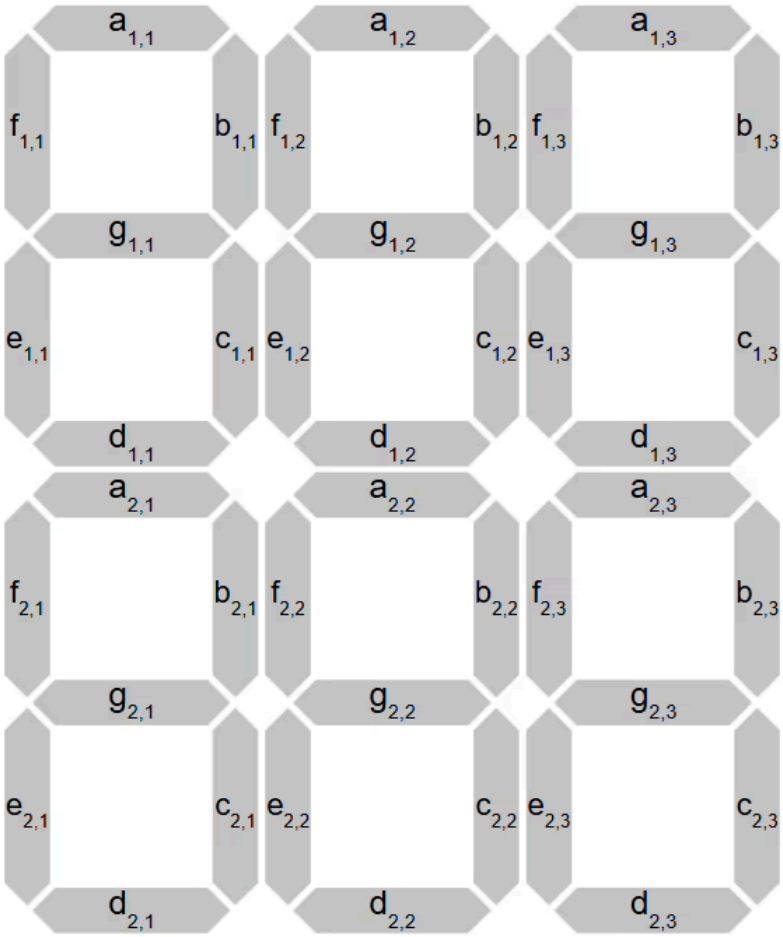
7-세그먼트 디스플레이에 세그먼트를 적절히 켜서 아래 그림처럼 0부터 9까지의 숫자 중 하나를 표시할 수 있다. 초록 세그먼트가 켜져 있는 세그먼트를 나타낸다.



알고리즘 학회 영과일은 각 칸이 7-세그먼트 디스플레이로 구성된 N행 M열 격자 형태의 전광판을 하나 보유하고 있다.

전광판의 x행 y열에 해당하는 디스플레이의 각 세그먼트는 각각 $a_{x,y}, b_{x,y}, \cdots, g_{x,y}$와 같이 나타낼 수 있다.

아래 그림은 2행 3열로 이루어진 전광판의 예시이다.



영과일의 성실한 학회원 혁준이는 이 전광판을 이용해 N행 M열의 숫자 데이터를 나타내려고 한다.

처음에, 이 전광판에 존재하는 모든 세그먼트는 꺼져있다.

세그먼트를 켜는 방법은 **직접 연결**과 **간접 연결**, **공유 연결** 세 가지가 존재한다.

- 세그먼트 위치를 나타내는 문자 p와 정수 x, y를 골라서 **직접 연결**하면 세그먼트 $p_{x,y}$가 켜진다. ($p \in \{a, b, c, d, e, f, g\}$; $1 \le x \le N$; $1 \le y \le M$)
- 켜진 세그먼트와 **인접한** 다른 세그먼트를 **간접 연결**하면 해당 세그먼트가 켜진다. **인접한** 세그먼트의 정의는 노트 탭을 참고한다.
- **직접 연결** 혹은 **간접 연결**된 세그먼트와 같은 디스플레이에 존재하는 세그먼트들은 **공유 연결**을 통해 켤 수 있다.

직접 연결을 한 번만 수행하여 N행 M열의 숫자 데이터를 모두 전광판에 나타낼 수 있는지 알아보자.

입력

첫 번째 줄에 전광판과 숫자 데이터의 크기를 나타내는 두 정수 N과 M이 공백으로 구분되어 주어진다. ($1 \le N, M \le 1\,000$)

두 번째 줄부터 N개의 줄에 걸쳐 숫자 데이터가 한 줄에 M개씩 공백 없이 주어진다. 모든 데이터는 0부터 9까지의 숫자 중 하나이다.

출력

직접 연결을 한 번만 수행하여 숫자 데이터를 모두 전광판에 나타낼 수 있으면 `YES` , 그렇지 않으면 `NO` 를 출력한다.

예제 입력 1 복사

```
2 2
00
01
```

예제 출력 1 복사

```
NO
```

예제 입력 2 복사

```
2 3
234
568
```

예제 출력 2 복사

```
YES
```

노트

두 세그먼트가 서로 긴 쪽 면을 맞대고 있다면 인접한 세그먼트라고 한다.

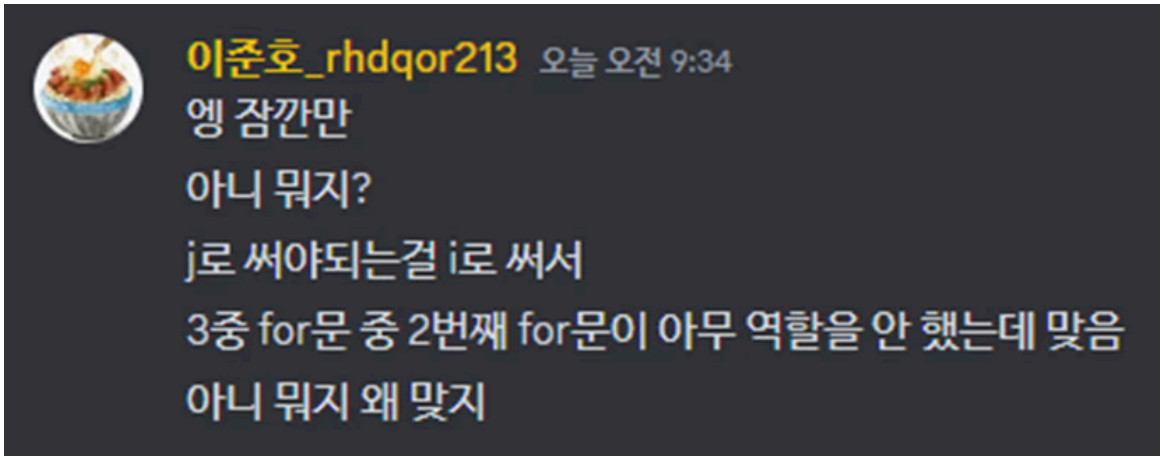
아래 그림에서 인접한 세그먼트끼리는 같은 색상으로 색칠되어 있다.



6681번 - 불필요한 for문

시간 제한	메모리 제한	제출	정답	맞은 사람	정답 비율
1 초	1024 MB	0	0	0	0.000%

문제



준호는 최근 백준 문제를 풀던 중 실수로 변수명을 잘못 써서 코드에 불필요한 for문이 있음에도 맞았습니다를 받았다!

처음 겪어보는 상황에 준호는 혹시 전에 제출했던 코드에도 불필요한 for문이 있지 않았을까? 생각하고 과거에 제출했던 코드들을 살펴보기로 했다.

입력으로 주어질 코드의 형식은 다음과 같다.

- 코드는 한 줄 이상의 for문과 마지막 한 줄의 연산식으로 이루어져 있다.
- for문을 구성하는 각 줄은 `for (int [변수명] = 1; [변수명] < 9; [변수명]++)` 의 형식으로 주어진다. 이는 [변수명]이라는 변수를 선언한 것이며, 세 [변수명]은 동일하다.
- 변수명은 알파벳 한 글자이며, 대소문자를 구분한다. 즉, n과 N은 다른 변수이다.
- 연산식은 알파벳, 숫자, 공백, 특수문자(아스키코드 32-126)로 이루어져 있고, 공백으로 시작/끝나지 않는 길이 1 이상 20 이하의 문자열이다.
- 연산식의 알파벳들은 반드시 변수명만을 의미하며 알파벳이 연속으로 주어지지 않는다.
- 연산식은 실제 프로그래밍 코드 형식과는 무관하다. 또한 선언되지 않은 변수가 사용될 수 있다.

여기서 불필요한 for문이란 다음 두 가지 조건 중 하나라도 만족하는 for문을 말한다.

- 해당 for문에서 선언한 변수가 내부의 다른 for문에서 다시 선언된 경우
- 해당 for문에서 선언한 변수가 연산식에서 전혀 사용되지 않은 경우

준호의 코드가 주어지면 불필요한 for문들을 모두 제거한 코드를 출력하자!

입력

여러 줄에 걸쳐 준호의 코드가 주어진다. 코드는 최대 500줄이다.

두 번째 줄부터는 코드의 앞에 한 줄에 하나씩 공백(space) 문자가 추가된다.

출력

여러 줄에 걸쳐 준호의 코드에서 불필요한 for문에 해당하는 행을 모두 지운 코드를 출력한다.

두 번째 줄부터는 코드의 앞에 한 줄에 하나씩 공백(space) 문자를 추가한다.

예제 입력 1 복사

```
for (int i = 1; i < 9; i++)
 for (int j = 1; j < 9; j++)
  x += i;
```

예제 출력 1 복사

```
for (int i = 1; i < 9; i++)
 x += i;
```

예제 입력 2 복사

```
for (int i = 1; i < 9; i++)
 for (int j = 1; j < 9; j++)
  x += i + j;
```

예제 출력 2 복사

```
for (int i = 1; i < 9; i++)
 for (int j = 1; j < 9; j++)
  x += i + j;
```

예제 입력 3 복사

```
for (int i = 1; i < 9; i++)
 for (int j = 1; j < 9; j++)
  for (int i = 1; i < 9; i++)
   -i+L)(9j
```

예제 출력 3 복사

```
for (int j = 1; j < 9; j++)
 for (int i = 1; i < 9; i++)
  -i+L)(9j
```

6692번 - 영일 마을에 살고 있는 엄은 친구의 집에 가고 싶다

시간 제한	메모리 제한	제출	정답	맞은 사람	정답 비율
2 초	1024 MB	0	0	0	0.000%

문제

영일 마을에는 엄과 N명의 친구들이 살고 있다. 영일 마을은 $N+1$개의 집이 M개의 도로로 연결되어 있으며, 엄의 집은 1번, 친구들의 집은 각각 2번부터 $N+1$번까지의 번호가 매겨져 있다. 엄의 집에서 모든 친구들의 집에 방문하는 경로가 있음이 보장된다.

모처럼 여유로운 엄은 자신의 집에서 출발하여 모든 친구들의 집을 방문하려 했지만, K명의 친구들이 집 문을 잠그고 여행을 떠나버렸다. 이때, 문이 잠긴 집과 연결된 도로는 모두 사용할 수 없다.

K명의 친구들이 집 문을 잠그고 여행을 떠났을 때, 엄이 방문할 수 있는 친구 집의 수를 구하여라.

입력

첫 번째 줄에 친구의 수 N, 도로의 수 M, 여행을 떠난 친구의 수 K가 주어진다. $(1 \leq N \leq 5\,000;\ N \leq M \leq \min(\frac{N(N+1)}{2}, 10\,000);\ 1 \leq K \leq N)$

두 번째 줄부터 M개의 줄에 도로의 정보 u, v가 공백으로 구분되어 주어진다. 이는 u번 집과 v번 집이 양방향 도로로 연결되어 있다는 것을 의미한다. 같은 도로의 정보는 주어지지 않는다. $(1 \leq u, v \leq N+1; u \neq v)$

마지막 줄에 여행을 떠난 K명의 친구들의 집 번호가 중복 없이 공백으로 구분되어 주어진다. 엄의 집 번호는 주어지지 않는다.

입력으로 주어지는 모든 수는 정수이다.

출력

엄이 방문할 수 있는 친구 집의 수를 출력한다.

예제 입력 1 복사

```
6 9 2
1 3
1 5
1 6
2 5
2 6
3 4
3 5
7 6
2 7
5 6
```

예제 출력 1 복사

```
2
```

6682번 - OR이 아니면? XOR

시간 제한	메모리 제한	제출	정답	맞은 사람	정답 비율
1 초	1024 MB	0	0	0	0.000%

문제

길이가 N인 정수 수열 A와 정수 M, K가 주어질 때, 아래 조건을 만족하는 (i, j) 쌍의 개수를 구하시오.

1. $A_i \oplus A_j = K$
2. $j - i \leq M$ $(i < j)$

A_i는 A의 i번째 원소를 의미한다.

입력

첫 번째 줄에 수열의 길이 N, M, K가 주어진다. $(2 \leq N \leq 10^6; 1 \leq M \leq N - 1; 0 \leq K \leq 2^{17} - 1)$

두 번째 줄에 수열 A의 원소 A_i가 공백으로 구분되어 N개 주어진다. $(0 \leq A_i \leq 100\,000)$

출력

첫 번째 줄에 조건을 만족하는 (i, j) 쌍의 개수를 출력한다.

예제 입력 1 복사

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

예제 출력 1 복사

```
1
```

예제 입력 2 복사

```
10 5 7
1 7 3 6 0 2 5 6 4 8
```

예제 출력 2 복사

```
3
```

예제 입력 3 복사

```
4 3 7
2 5 2 5
```

예제 출력 3 복사

```
4
```

예제 입력 4 복사

```
9 4 3
5 7 9 11 13 15 17 19 21
```

예제 출력 4 복사

```
0
```

힌트

음이 아닌 두 정수 A, B의 배타적 논리합 $A \oplus B$는 다음과 같이 정의된다.

이진법으로 생각했을 때, A의 2^k의 자릿수와 B의 2^k의 자릿수가 서로 다르면 $A \oplus B$의 2^k의 자릿수가 1이고, 같으면 $A \oplus B$의 2^k의 자릿수가 0이다. (단, $k \geq 0$)

예를 들어 $12 \oplus 10$은 $12 = 1100_{(2)}$, $10 = 1010_{(2)}$이므로 $1100_{(2)} \oplus 1010_{(2)} = 0110_{(2)} = 6$이다.

6689번 - 투명 스프레이

시간 제한	메모리 제한	제출	정답	맞은 사람	정답 비율
1 초	1024 MB	0	0	0	0.000%

문제

월급 루팡 정민이는 일을 하지 않기 위해 사람들을 피해서 퇴근하려 한다.

건물은 $N \times M$ 크기의 격자로 구성되어 있고, 격자 (i, j)에서 사람을 만날 위험도는 F_{ij}이다. 정민이는 좌측 상단 $(1, 1)$에서 우측 하단 (N, M)으로 이동해야 한다.

이동은 상하좌우 인접 칸으로만 가능하며, 만약 이동하려는 칸의 위험도가 X 초과라면 해당 칸을 지나기 위해선 투명 스프레이가 필요하다. 시작 칸 $(1, 1)$의 위험도는 항상 0이다.

정민이는 칼퇴를 위해 투명 스프레이를 K개 구비해 두었다. 투명 스프레이를 하나 사용하여 한 칸의 위험도를 무시하고 지나갈 수 있고, 스프레이가 적용된 칸의 효과는 재방문해도 유지된다.

퇴근 경로 상에서 투명 스프레이 사용 횟수가 K 이하가 되도록 하는 최소의 위험도 X를 구하라. 단, 위험도는 음이 아닌 정수이다.

입력

첫 번째 줄에 세 정수 N, M, K가 주어진다. ($2 \leq N, M \leq 700, 0 \leq K \leq 1\,400$)

그 다음 N개의 줄에 걸쳐, 각 줄에 M개의 정수 F_{ij}가 주어진다. ($0 \leq F_{ij} \leq 10^9$)

출력

첫 번째 줄에 문제의 정답을 출력한다. X의 값에 상관없이 퇴근이 불가능한 경우에는 -1을 출력한다.

예제 입력 1 복사

```
3 3 1
0 2 3
3 2 1
4 5 1
```

예제 출력 1 복사

```
2
```

예제 입력 2 복사

```
3 3 1
0 1 100
100 100 1
1 1 1
```

예제 출력 2 복사

```
1
```

6683번 - 점심 나가서 먹기

시간 제한	메모리 제한	제출	정답	맞은 사람	정답 비율
1 초	1024 MB	0	0	0	0.000%

문제

매일같이 알고리즘 대회 준비와 학교 수업으로 바쁜 준호는 하루 종일 머리를 혹사당하고 있다.

가혹한 일정에 정신이 나가버린 준호는... 점심을 나가서 먹기로 했다!

준호는 N개의 건물들과 M개의 도로들로 이루어진 도시에 살고 있다. 도시의 건물들은 1번부터 N번까지 번호가 붙어 있으며, 도시 곳곳에는 공사 중인 도로도 있어 어떤 건물들은 방문할 수 없는 경우도 있다. 도시에는 수많은 식당과 카페들이 있지만, 가난한 준호는 방문할 수 있는 곳 중 가장 싼 식당과 가장 싼 카페를 찾아 점심을 해결하고 돌아오기로 했다. (물론 가장 싼 식당과 카페가 같은 건물에 있을 수도 있다.)

점심을 빠르게 먹고 돌아오고 싶은 준호는 식당과 카페를 들르는 순서는 상관없이, 최소 거리로 돌아오고 싶어 한다.

준호가 집(1번 건물)에서 출발하여 도달할 수 있는 가장 싼 식당과 카페를 들른 후, 다시 집으로 돌아오는 최소 거리를 구해보자.

입력

첫 번째 줄에 준호의 집을 포함한 건물의 수 $N(2 \le N \le 100\,000)$과 건물 간의 도로 개수 $M(1 \le M \le 200\,000)$이 주어진다.

두 번째 줄에 i번 건물 식당들 중 최소 가격 $x_i(0 \le x_i \le 10^9)$가 N개의 정수로 주어진다. 각각의 가격은 모두 다르며, 0일 경우 해당 건물에는 식당이 없다.

세 번째 줄에 j번 건물 카페들 중 최소 가격 $y_j(0 \le y_j \le 10^9)$가 N개의 정수로 주어진다. 각각의 가격은 모두 다르며, 0일 경우 해당 건물에는 카페가 없다.

다음 M개의 줄에는 세 개의 정수 $u, v, w(1 \le u, v \le N; 1 \le w \le 10\,000)$가 주어지며, 이는 u번 건물과 v번 건물이 거리 w인 양방향 도로로 연결되어 있음을 의미한다.

u와 v는 같을 수 있다.

준호의 집에는 식당과 카페가 없으며, 준호의 집에서 적어도 하나의 식당과 카페에 도달할 수 있음이 보장된다.

출력

집(1번 건물)에서 출발하여 도달할 수 있는 가장 싼 식당과 카페를 들렀다가 다시 집으로 돌아오는 최소 거리를 출력한다.

예제 입력 1 복사

```
6 7
0 5000 0 7000 0 0
0 0 0 0 3000 4000
1 2 4
1 3 2
2 4 5
3 4 3
4 5 1
5 6 6
3 6 7
```

예제 출력 1 복사

```
16
```

6691번 - 지형 평탄화 탐색기

시간 제한	메모리 제한	제출	정답	맞은 사람	정답 비율
5 초	1024 MB	0	0	0	0.000%

문제

(주)영일소프트는 지형의 고도차가 심한 지역에 도시 기반 시설을 구축하기 위한 건설 시뮬레이션 시스템을 개발하고 있다.

이 시스템에서는 두 가지 핵심 정보가 사용된다.

- 지도 : 해당 지역의 지형에 대한 고도를 기록한 $N \times M$ 크기의 격자판으로, 각 지점의 현재 고도 정보를 가지고 있다.
- 작업 계획도 : $r \times c$ 크기의 격자판으로, 각 칸에는 얼마만큼의 높이를 줄여야 하는지 적혀있다.

지도에서 $r \times c$ 크기의 격자 영역을 선택하여 작업 계획도와 정확히 겹치도록 한 후, 각 지점에 대해 지도에 적힌 고도에서 작업 계획도에 적힌 값을 빼보았을 때, 해당 격자 영역 내 모든 지점의 고도가 동일해진다면 해당 위치는 평탄화가 가능한 부지로 판단한다. 단, 작업 계획도를 회전하거나 변형할 수 없으며, 주어진 형태 그대로 적용해야 한다. 음수의 고도로도 평탄화가 가능함에 유의한다.

사용자는 다양한 작업 계획도를 생성하거나, 지도에서 특정 지점의 고도를 직접 수정해 가며 평탄화가 가능한 부지를 탐색한다.
이러한 작업을 반복하면서 도시 건설이 가능한 부지를 시뮬레이션하고, 최적의 부지 후보를 확보하는 것이 목표다.

프로그램은 다음과 같은 Q개의 쿼리를 처리해야 한다.

- $1\ x\ y\ h$
  - 지도에서 x번째 행, y번째 열의 고도를 h로 수정한다. $(1 \le h \le 5)$
- $2\ r\ c\ a_{1,1}\ a_{1,2} \dots\ a_{r,c}$
  - $r \times c$ 크기의 작업계획도가 주어진다. i번째 행, j번째 열에 해당하는 칸에는 줄여야 하는 고도 값 $a_{i,j}$가 기록되어 있다. 이 계획도를 지도 위의 모든 가능한 위치에 적용해 보면서, 해당 부지가 평탄화 가능한지를 확인한다. 그 결과로, 평탄화가 가능한 부지의 총 개수를 출력한다.
  $(0 \le a_{i,j} \le 5; 1 \le r \le \min(9, N); 1 \le c \le \min(9, M);\ 1 \le r \times c \le 9)$

입력

첫 번째 줄에 지도의 행의 개수와 열의 개수를 나타내는 두 정수 N과 M이 공백으로 구분되어 주어진다. $(1 \le N, M \le 500)$

두 번째 줄부터 N개의 줄에 걸쳐 지도의 고도 정보가 주어진다. 각 줄은 한 행에 해당하며, M개의 정수가 공백으로 구분되어 주어진다. 이 정수들은 해당 행의 첫 번째 열부터 M번째 열까지의 해발 고도를 의미하며, 1 이상 5 이하의 정수이다.

그다음 줄에는 쿼리의 개수를 나타내는 정수 Q가 주어진다. 2번 쿼리는 최소 한 번 이상 주어진다. $(1 \le Q \le 100\,000)$

이후 Q개의 줄에 걸쳐 한 줄에 하나씩 쿼리가 주어진다.

출력

2번 쿼리가 주어질 때마다 그 결과를 한 줄에 하나씩 출력한다.

예제 입력 1 복사

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

예제 출력 1 복사

```
5
3
```

6687번 - 괄호 붙이기

시간 제한	메모리 제한	제출	정답	맞은 사람	정답 비율
1 초	1024 MB	0	0	0	0.000%

문제

올바른 괄호 문자열은 다음과 같이 정의한다.

1. 빈 문자열은 올바른 괄호 문자열이다.
2. S 가 올바른 괄호 문자열일 때, (S) 도 올바른 괄호 문자열이다.
3. S 와 T 가 올바른 괄호 문자열이라면, ST 도 올바른 괄호 문자열이다.

예를 들어, ()(), (()), ((((()))(()))) 등은 올바른 괄호 문자열이고, (((),)(,))))()()((등은 그렇지 않다.

(,) 로 이루어진 문자열 A, B가 주어진다.

A를 길이 1 이상의 연속된 두 문자열 A_1, A_2로 나눈 후 A_1, B, A_2를 이어붙여 새롭게 $A_1 + B + A_2$라는 문자열을 만들었을 때,

올바른 괄호 문자열이 되도록 A_1, A_2를 나눌 수 있다면 1, 아니면 0을 출력한다.

입력

첫 번째 줄과 두 번째 줄에 괄호로 이루어진 문자열 A, B가 주어진다. $(2 \leq |A| \leq 200\,000; 1 \leq |B| \leq 200\,000)$

출력

$A_1 + B + A_2$가 올바른 괄호 문자열이 되도록 A를 나누는 방법이 존재하면 1, 아니면 0을 출력한다.

예제 입력 1 복사

```
((
))
```

예제 출력 1 복사

```
0
```

예제 입력 2 복사

```
(()
)
```

예제 출력 2 복사

```
1
```