Поледица

Както всяка зима и тази Иванчо и приятели искат да излязат на вън да се попързалят на леда. Но за съжаление днес е прекалено топло и част от леда се е разтопил. Те решават да разделят пързалката на квадратни зони. Всяка зона може да бъде заледена или незаледена. Така си поставят за цел да намерят как с най-малко стъпвания на незаледени зони може да се стигне от зона **A** до зона **B**. Когато са попаднали на незаледена зона, могат да си изберат дали да продължат в една от четирите посоки - напред, назад, наляво или надясно, като при стъпване на заледена зона продължават да се движат в същатата посока докато не достигнат друга незаледена зона. Ако случайно излязат извън пързалката играта приключва.

Иванчо иска да бъде пръв, за това моли вас за помощ, като напишете програма **ice**, която по зададени зони и координати на **A** и **B**, да намира с колко най-малко стъпвания на незаледени зони може да се стигне от зона **A** до зона **B**.

**Вход:** На първия ред на входния файл **ice.in** е записано едно цяло положително число **N**, което е дължината и ширината на пързалката (NxN зони). На всеки от следващите **N** реда са записани **N** на брой единци и нули, като единиците оказват къде има лед, а нулите – къде няма. На следващите два реда са записани **XA**, **YA** и **XB**, **YB** – координатите на зона **A** и на зона **B**.

**Забележка: A** и **B** са винаги незаледени. **B** не се брои към крайния резултат, но **A** – не. Не може да се променя посоката по време на пързаляне. Индексацията на координатите започва от 1.

**Изход:** Изходният файл **ice.out** трябва да съдържа едно цяло неотрицателно число – търсеният от Иванчо брой или -1, ако не съществува път от началото до края.

**Ограничения:**

1 <= **N** <= 1000

1 <= **x, y** <= **N**

**Ограничение за време**: 1 сек

**Ограничение за памет**: 256 MB

Предварителни тестове: 4

Финални тестове: 10

**Примерен тест:**

|  |  |
| --- | --- |
| **ice.in** | **ice.out** |
| 5  0 0 0 0 0  0 1 1 1 0  0 0 1 0 0  0 0 0 1 0  0 0 0 0 0  1 1  5 5 | 5 |

**Обяснение на изхода**:

Има две възможности да се стигне от 1, 1 до 5, 5 по зададените правила:

1 1

1 2

5 2

5 3

5 4

5 5

и

1 1

2 1

3 1

3 4

4 5

5 5