# Упражнения: Многомерни масиви

Тествайте решението в Judge: <https://judge.softuni.bg/Contests/3173/Multidimentional-Arrays>

## Играта на коня

Шахът е най-старата игра, но все още е популярна. В задачата ще използваме само една фигура - **Конят.**

Конят се движи до **най-близкото** поле, **но не на същия ред, колона** или **диагонал**. (Има два вида движение. Първото е да се премести две хоризонтално и след това едно квадратче вертикално. Другото е едно квадратче хоризонтално и след това две квадратчета вертикално - т. е. в модел **"L"**.)

Играта на коня се играе на дъска с измерение NxN и много коне: **0 <= K <= N2**.

Ще получите дъска със символи. **'K'** означава, че на това място има кон. '**0'** означава е клетката е празна. Вашата задача е да премахнете минималния брой коне, така че да няма останали коне за атака на другия кон.

### Вход

* На първия ред ще получите числото **N** - размера на дъската.
* На следващите N реда ще получите низ от **'K'** или **'0'**.

### Изход

Принтирайте минималния брой коне, който трябва да се премахнат

### Бележки

* Размерът на дъската ще бъде от 0 до 3

#### Примери

|  |  |
| --- | --- |
| **Вход** | **Изход** |
| 5  0K0K0  K000K  00K00  K000K  0K0K0 | 1 |
| 2  KK  KK | 0 |
| 8  0K0KKK00  0K00KKKK  00K0000K  KKKKKK0K  K0K0000K  KK00000K  00K0K000  000K00KK | 12 |

## \*Бомба

Ще ви бъде дадена квадратна матрица от числа. Всяко число ще бъде разделено **с интервал**, а всяка редица на **нов ред**. На последния ред ще получите **индекси** - **координати** на няколко клетки, разделени **с интервал** в следния формат: **редица1,колона1 редица2,колона2 редица3,колона3...**

На тези клетки има **бомби**. Трябва да минете през **всички бомби**, както са дадени подред. Когато бомбата **експлодира**, нанася щети, **равни на стойността** на бомбата, на всички клетки около нея (във всички посоки и диагонали). Една бомба **не може** да се взриви **повече от един път**. След експлозията получава стойност **0**. Когато клетката е **по-малка или равна** на 0, тя **умира**. Умрелите клетки не могат **експлодират**.

Вие трябва да отпечатате **броя на живите клетки** и **тяхната сума**. След това **принтирайте всички клетки** (включително умрелите).

### Вход

* На първия ред ще получите цяло число N - размерът на квадратната матрица
* На следващите N реда ще получите стойностите за всяка редица - N числа, разделени с интервал
* На последния ред ще получите координати на клетките с бомби

### Изход

* На първия ред отпечатайте броя на оцелелите клетки в следния формат:

“**Alive cells: {aliveCells}**”

* На втория ред отпечатайте сумата от оцелелите числа

“**Sum: {sumOfCells}**”

* Накрая отпечатайте матрицата. Клетките трябва да бъдат **разделени с интервал**.

### Бележки

* Размерът на матрицата ще бъде в обхвата **[0…1000].**
* Координатите на бомбата **винаги** ще бъдат в матрицата.
* Стойностите на бомбата винаги ще са **по-големи от 0.**
* Числата в матрицата ще бъдат в обхвата **[1…10000].**

### Примери

|  |  |  |
| --- | --- | --- |
| **Вход** | **Изход** | **Обяснения** |
| 4  8 3 2 5  6 4 7 9  9 9 3 6  6 8 1 2  1,2 2,1 2,0 | Alive cells: 3  Sum: 12  8 -4 -5 -2  -3 -3 0 2  0 0 -4 -1  -3 -1 -1 2 | Първо бомбата със стойност **7** ще експлодира и ще намали стойностите на клетките около нея. След това бомбата с координати **2,1** и стойност **2** (първоначално 9-7) ще експлодира и ще намали съседните си клетки. В крайна сметка бомбата с координати 2.0 и стойност **7** (първоначално 9-2) ще избухне. След това трябва да отпечатате броя на живите клетки, който е 3, а сумата им е 12. Отпечатайте матрицата след експлозиите. |
| 3  7 8 4  3 1 5  6 4 9  0,2 1,0 2,2 | Alive cells: 3  Sum: 8  4 1 0  0 -3 -8  3 -8 0 | |

## \*Гадни зайци

Разглеждате си GitHub, откривате някаква стара js игра. Става въпрос за гадни зайци, който се размножават много бързо. Има и играч, който трябва да избягва от тяхната бърлога. Много ви допада тази игра и решавате да я направите на C#, защото това е езикът, който сте избрали. Алгоритъмът на играта трябва да реши дали играчът ще успее да избяга.

Първо ще трябва да получите числата **N** и **M**, които са колоните и редовете на бърлога. След това ще получите **N** низове, като съдържат символите **".", "B", "P".** Зайците са маркирани с **"B"**. **Играчът** е маркиран с **"P"**. **Всичко друго** е запълнено с **"."**. От тези символи е изградено първоначалното състояние на бърлогата. Винаги ще има само един играч. След това ще получите **низ** с команди като **LLRRUUDD -** всеки символ означава на къде да се **премести** играча (L - наляво, R - надясно, U - нагоре, D - надолу).

**След всяка стъпка**, зайците ще опитват да го **спрат** като се разпространяват наляво, надясно, нагоре, надолу (съседната клетка от '.' се **променя** на 'B'). Ако играчът се **премести** на клетка, където има **заек** или някой заек **хване** играчът, то той **умира**. Ако **излезе** без да бъде хванат **печели** играта.

Когато играчът **спечели или умре**, играта свършва. Няма да има такива примери, в които играчът да няма ходове преди да умре или да избяга.

Накрая трябва да отпечатате всяка редица на нов ред. На последния ред отпечатайте **“dead: {ред} {колона}”** или **“won: {ред} {колона}”.** Редът и колоната са координатите на клетката на играчът ако е умрял или последната клетка, на която е бил преди да избяга.

### Вход

* На първият ред ще получите числата **N** и **M** - номерата на **редовете** и **колоните** на бърлогата
* На следващите N реда ще получите редовете като низ. Низът съдържа само “.”, “B”, “P”. Всички низове ще имат еднаква големина. Ще има само един символ от “P”
* На последният ред ще получите посоките като низ, който съдържа “R”, “L”, “U”, “D”

### Изход

* На първите N реда отпечатайте крайния резултат от бърлогата
* На последния ред отпечатайте – “won:” or “dead:” + {row} {col}

### Бележки

* Директорията е низ в обхвата [1...20]

### Примери

|  |  |  |  |
| --- | --- | --- | --- |
| **Вход** | **Изход** | **Вход** | **Изход** |
| 5 8  .......B  ...B....  ....B..B  ........  ..P.....  ULLL | BBBBBBBB  BBBBBBBB  BBBBBBBB  .BBBBBBB  ..BBBBBB  won: 3 0 | 4 5  .....  .....  .B...  ...P.  LLLLLLLL | .B...  BBB..  BBBB.  BBB..  dead: 3 1 |