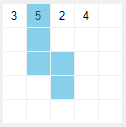
Кръг III. Игра

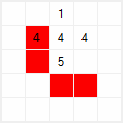
След като Том Сойер измами Бен Роджърс да боядиса оградата вместо него, той реши да му предложи реванш в следната игра. Двамата ще играят на квадратна дъска с големина **N**x**N**. Всяко квадратче от дъската може или да е празно, или да съдържа число от **1** до **C**. Правилата на играта са следните:

1. На всеки ход някой от играчите боядисва произволен брой полета *(за прегледност по-долу ще боядисваме невалидните ходове в червено)*.
2. Всички боядисани в един ход квадрати трябва да бъдат свързани помежду си, т.е. да имат обща стена.

Валиден ход:

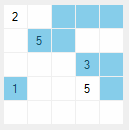


Невалиден ход:

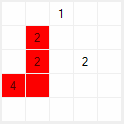


1. Множеството от боядисаните квадрати в един ход трябва да съдържа точно един квадрат с число.

Валиден ход:

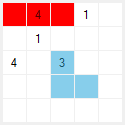


Невалиден ход:



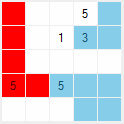
1. Броят на квадратите в множеството трябва да е равен точно на числото в него.

Невалиден ход:



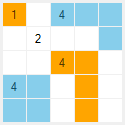
1. Две множества не трябва да се допират едно с друго.

Невалиден ход:



1. Играчите нямат право да боядисват вече боядисани полета.
2. Губи този, който не може да направи ход.

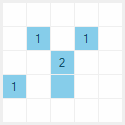
Пример:



(в случая, ако ходовете на първия играч са тези в синьо, а тези на втория в оранжево, то губещ е синият, тъй като той не може да направи валиден ход след последния оранжев)

1. В случай, че всички числа са в някаква област, то играта свършва с равенство.

Пример:



Помогнете на Бен този път да победи Том, като напишете програма, която да играе оптимално. На всеки ход тя ще бъде извиквана на ново, като от входния файл **game.in** ще трябва да вземе текущата дъска, след което във файла **game.out** да запише поредния ход. Дъската ще бъде дефинирана по следния начин: размерът на дъската ще бъде отбелязван с числото **N**. Дъската ще може да бъде представена като матрица с големина **NxN**. Ако полето е празно, то елементът от матрицата ще бъде със стойност „#“, ако е оцветено с “B”, а ако съдържа число - с неговата стойност.

# Оценяване:

Програмата на всеки ще бъде тествана срещу тази на всеки от останалите. Ако вашето решение победи това на съперника Ви, печелите 3 точки, а в случай на равенство и двамата взимате по 1.5. Ако програмата Ви изведе невалиден ход, получавате 0 точки, а съперникът Ви печели. Всички ще играят на всяка от дъските по два пъти, разменяйки се кой да е първи и кой да бъде втори.

***Забележка:*** *При изпращане на решение по време на кръга ще получавате информация единствено за това дали първият ход, който направи вашата програма е валиден. Финалното тестване ще се осъществи след края на кръга.*

# Вход:

От първия ред от входния файл **game.in** се въвежда числото **N** – големината на дъската. Следват **N** реда с по **N** елемента на матрицата, отделени с празно място помежду си, описващи дъската в текущото й състояние.

# Изход:

На първия ред от изходния файл **game.out** програмата трябва да изведе едно число, отговарящо за броя квадрати, които ще оцвети в своя ход.

Следват двойки числа **x** и **y**, всяка на нов ред, отговарящи за позицията, която трябва да се боядиса. **X** отговаря за ред, а **y** за колона, като елементът в горния ляв ъгъл се намира на координати **(x, y) =** **(0, 0)**.

# Ограничения:

*Ограничението за време на изпълнение на програмата е 1 секундa.*

# Пример:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ход | | | | |
| 1. | **Първи - game.in:** | **Първи – game.out:** | **Описание:** | |
| 5  # # # # #  # # 3 1 #  # # # # #  2 1 # 5 #  # # # # # | 5  3 3  3 4  4 2  4 3  4 4 |  | |
| 2. | **Описание:** | | **Втори - game.in:** | **Втори - game.out:** |
|  | | 5  # # # # #  # # 3 1 #  # # # # #  2 1 # B B  # # B B B | 1  1 3 |
| 3. | **Първи - game.in:** | **Първи – game.out:** | **Описание:** | |
| 5  # # # # #  # # 3 B #  # # # # #  2 1 # B B  # # B B B | 2  3 0  4 0 |  | |
| КРАЙ | Няма повече възможни ходове, т.е. **първият** играч печели. | | | |
|

# Генериране на тестовете:

Ще бъдат генерирани три групи от тестове, всяка с по четири теста. В първата група големината на дъската ще бъде **N** = 5, броят центрове на концентрация на полетата **S** = 1 и максималната големина на числото в някое поле ще бъде **C** Є [0, 5]. Във втората **N** = 15, **S** = 3, **C** Є [5, 10]. А в третата **N** = 30, **S** = 5, **C** Є [10, 20]. По време на кръга ще получите по една дъска от всеки вид, с която можете да тествате своите решения. След края на кръга тестването ще се осъществи с останалите три дъски от всяка група. Разположението на полетата ще бъде генерирано с помощта на функция за нормално разпределение ([normal distribution](http://en.wikipedia.org/wiki/Normal_distribution)). Генераторът може да бъде описан със следния псевдокод:

|  |
| --- |
| **N** // Board size  **A** := **N** \* **N** // All elements  **S** // Number of centers  **a**, **b** // C Є [a, b]  **Centers** := **new int** [**S**, 2] // coordinates of the centers  **Board** := **new int** [**N**, **N**] // board values  **sigma** := 0.3, **B** := 0.6 // some coeficients  **E** = *Euler's constant*  **function** **f** ( **int** **d** ): // normal distribution formula with µ=0  **return** ( **Pow** ( **E**, -**double**(**d**)/(2.0 \* **Sqrt**(**A**) \* **sigma** \* **sigma**) ) / (**Sqrt** (2.0 \* **PI** ) \* **sigma**) ) \* **B**  **SetRandomSeed**(**Seed**)  **for** **i** from 0 to 5:// placing the centers at random integer positions  **Centers**[**i**, 0] = **Rand**(0, **N - 1**);  **Centers**[**i**, 1] = **Rand**(0, **N - 1**);  **for** **i** from 0 to **N**:  **for** **j** from 0 to **N**:  // find the distance to the closest center = d  **d** := **INF** // infinity  **for** **k** from **0** to **S**:  **d** := **Min** { **d**, **Abs**(**i** - **Centers**[**k**,0]) + **Abs**(**j** - **Centers**[**k**,1]) }  // ------------------------------------------------  // pick some random number in the interval [0, 1]  **p** := **Rand**(0, 1);  **if** **p** < **f**(**d**): // normal distribution  **Board**[**i**, **j**] := **Rand**(**a**, **b + 1**) // put some random number from [a, b] at position (i, j) |