

Игра с плочки

Барчин и Чарос играят игра на таблица от $2N \times 2M$ единични квадратни клетки. Редовете са номерирани от 0 до $2N - 1$ от горе надолу, а колоните са номерирани от 0 до $2M - 1$ от ляво надясно. За $0 \leq i < 2N$ и $0 \leq j < 2M$, обозначаваме клетката в ред i и колона j с (i, j) .

Барчин дава на Чарос $N \cdot M$ **блокчета**, едно по едно. Всяко блокче е 2×2 квадрат, състоящ се от четири 1×1 **плочки**. Барчин е оцветил всяка плочка от блокчето в черно или бяло, като гарантира, че **поне една плочка е бяла**.

Чарос трябва да постави всяко блокче върху таблицата **веднага след получаването му**, без да знае какви блокчета ще получи по-късно. Блокчетата не могат да бъдат въртени. Всяко блокче трябва да бъде поставено изцяло в таблицата, покривайки точно четири клетки от нея. Освен това, горната лява плочка на всяко блокче трябва да покрива клетка, чиито ред и колона са с четни номера. Всяка клетка от таблицата може да бъде покрита от най-много едно блокче.

Барчин печели играта, ако след поставянето на някое блокче съществува 2×2 квадрат, покрит с четири черни плочки. Формално, ако клетките (a, b) , $(a + 1, b)$, $(a, b + 1)$, $(a + 1, b + 1)$ са покрити с черни плочки за някои $0 \leq a < 2N - 1$ и $0 \leq b < 2M - 1$, тогава Барчин печели. Индексите a и b не е необходимо да са четни.

Чарос печели, ако постави всичките $N \cdot M$ блокчета, без Барчин да спечели. Обърнете внимание, че поставянето на $N \cdot M$ блокчета ще покрие изцяло таблицата.

Вашата задача е да имплементирате стратегия за Чарос, така че да спечели играта. Може да се докаже, че при дадените ограничения Чарос винаги може да разположи блокчетата така, че да гарантира победа, независимо от оцветяването на блокчетата, които получава по-късно.

Детайли по имплементацията

Трябва да имплементирате две функции:

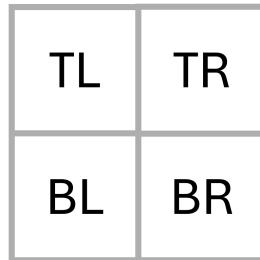
```
void init(int N, int M)
```

- N : половината от броя на редовете в таблицата.
- M : половината от броя на колоните в таблицата.

- Функцията се извиква точно веднъж на тест, в началото на изпълнението на Вашата програма.

```
std::pair<int, int> receive_block(int TL, int TR, int BL, int BR)
```

- TL, TR, BL, BR : цветовете, съответно на горния ляв, горния десен, долния ляв и долния десен ъгъл на текущото блокче, както е показано на картинката по-долу. Всяка стойност е или 0 (бяло), или 1 (черно).
- Тази процедура се извиква точно $N \cdot M$ пъти на тест, след първоначалното извикване на `init`.



Тази функция трябва да върне двойка цели числа (i, j) , където i е номера на реда, а j е номера на колоната на клетката, където трябва да се постави горната лява плочка на това блокче. Както i , така и j трябва да бъдат четни числа, а 2×2 квадрата, покрит от блокчето, не трябва да се припокрива с никое от преди това поставените блокчета.

Ако `receive_block` някога върне двойка, която не отговаря на тези изисквания, или ако след поставянето на блокче, някой 2×2 квадрат бъде изцяло покрит с черни плочки, грейдърът незабавно ще прекрати програмата Ви и резултатът за теста ще бъде `Output isn't correct`.

Поведението на грейдъра **не е адаптивно**. Това означава, че последователността от блокчета, които Барчин дава на Чарос, е фиксирана преди извикването на `init`.

Ограничения

За всяко блокче, нека S е броят черни плочки сред неговите четири плочки, т.е. $S = TL + TR + BL + BR$.

- $1 \leq N, M \leq 100$
- $0 \leq S \leq 3$ за всяко блокче.

Подзадачи

Подзадача	Точки	Допълнителни ограничения
1	6	$S = 1$ за всяко блокче и $N = 2$.
2	16	$S = 3$ за всяко блокче. $N = M$, N е четно число и всяко от четирите възможни оцветявания на блокчета се появява точно $\frac{N^2}{4}$ пъти.
3	10	$S = 1$ за всяко блокче.
4	29	$S \leq 2$ за всяко блокче.
5	39	Няма допълнителни ограничения.

Пример

Да разгледаме игра с $N = 1$ и $M = 2$, така че таблицата има 2 реда и 4 колони. Грейдърът първо извиква:

```
init(1, 2)
```

Първоначално всички клетки са празни. Таблицата изглежда по следния начин:

	0	1	2	3
0				
1				

Има $N \cdot M = 2$ блокчета за поставяне. Да предположим, че Барчин е дал блокче с три черни плочки и една бяла плочка в горния десен ъгъл. Грейдърът извиква:

```
receive_block(1, 0, 1, 1)
```

Чарос решава да постави това блокче от лявата страна на таблицата, като връща $(0, 0)$.

Таблицата сега изглежда по следния начин:

	0	1	2	3
0				
1				

След това Барчин дава друго блокче с три черни плочки и една бяла плочка в горния ляв ъгъл:

```
receive_block(0, 1, 1, 1)
```

Единствената останала клетка с четен ред и четна колона, която може да бъде горен ляв ъгъл на 2×2 блокче е $(0, 2)$, така че Чарос връща $(0, 2)$. Крайната таблица изглежда по следния начин:

	0	1	2	3
0				
1				

Никой 2×2 квадрат не е изцяло покрит с черни плочки, така че Чарос е поставила успешно всички блокчета, без Барчин да спечели. Чарос печели играта.

Локален грейдър

Формат на входа:

```
N M
TL[0] TR[0] BL[0] BR[0]
TL[1] TR[1] BL[1] BR[1]
...
TL[NM-1] TR[NM-1] BL[NM-1] BR[NM-1]
```

Формат на изхода:

```
R[0] C[0]  
R[1] C[1]  
...  
R[NM-1] C[NM-1]
```

Тук $R[k]$ и $C[k]$ са двойката цели числа, върнати от k -тото извикване на `receive_block`.