ფილების დაგების თამაში

ბარჩინი და ქაროსი თამაშობენ თამაშს $2N \times 2M$ ზომის უჯრებიან დაფაზე. სტრიქონები გადანომრილია 0-დან $2N-1$-ის ჩათვლით ზემოდან ქვევით, ხოლო სვეტები 0-დან $2M-1$-ის ჩათვლით-მარცხნიდან მარჯვნივ. როდესაც $0 \le i < 2N$ და $0 \le j < 2M$, i-ური სტრიქონისა და j-ური სვეტის უჯრას აღვნიშნავთ (i, j)-ით,

ბარჩინი ქაროსს აძლევს $N \cdot M$ ცალ **ბლოკს**, ერთმანეთის მიყოლებით. თითოეული ბლოკი არის 2×2 ზომის **კვადრატი**, რომელიც შედგება ოთხი 1×1 ზომის **ფილისგან**. ბარჩინმა ბლოკის თითოეული ფილა შეღება შავად ან თეთრად, ისე, რომ, **სულ მცირე ერთი ფილა თეთრია**.

ქაროსმა თითოეული ბლოკი **მიღებისთანავე** უნდა განათავსოს დაფაზე, მოგვიანებით მისაღები ბლოკების ცოდნის გარეშე. ბლოკების მოტრიალება არ შეიძლება. თითოეული ბლოკი მთლიანად უნდა მოთავსდეს დაფის შიგნით და ზუსტად ოთხი უჯრა უნდა დაფაროს. გარდა ამისა, თითოეული ბლოკის ზედა მარცხენა უჯრამ უნდა დაფაროს ისეთი უჯრა, რომლის სტრიქონისა და სვეტის კოორდინატები ლუწი რიცხვებია. დაფის თითოეული უჯრა შეიძლება დაიფაროს მაქსიმუმ ერთი ბლოკით.

ბარჩინი თამაშს იგებს, თუ რომელიმე ბლოკის განთავსების შემდეგ არსებობს 2×2 ზომის კვადრატი, რომელიც ოთხი შავი ფილითაა დაფარული. ფორმალურად, თუ რომელიმე $0 \le a < 2N-1$-სა და $0 \le b < 2M-1$-ისთვის, უჯრებიდან (a, b), $(a+1, b)$, $(a, b+1)$, $(a+1, b+1)$ ყველა დაფარულია შავი ფილებით, მაშინ ბარჩინი იგებს. a და b არაა აუცილებელი ლუწი იყოს.

ქაროსი იგებს თუ ის ყველა $N \cdot M$ ბლოკს ისე განათავსებს, რომ ბარჩინმა ვერ მოიგოს. ყურადღება მიაქციეთ, რომ $N \cdot M$ ცალი ბლოკის განთავსება მთლიანად შეავსებს დაფას.

თქვენი ამოცანაა, შეიმუშავოთ სტრატეგია, რომლითაც ქაროსს თამაშის მოგება შეეძლება. დამტკიცებადია, რომ მოცემული შეზღუდვების პირობებში, ქაროსს ყოველთვის შეუძლია ბლოკების ისე განლაგება, რომ გარანტირებულად მოიგოს, მიუხედავად იმისა, თუ როგორ გაფერადდება მოგვიანებით მისაღები ბლოკები.

იმპლემენტაციის დეტალები

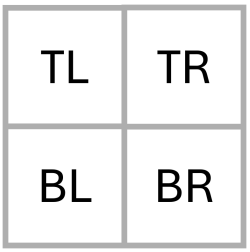
თქვენ უნდა დააიმპლემენტიროთ 2 პროცედურა:

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

- N: დაფის სტრიქონების რაოდენობის ნახევარი.
- M: დაფის სვეტების რაოდენობის ნახევარი.
- ეს პროცედურა გამოიძახება ზუსტად ერთხელ თითოეულ ტესტში, თქვენი პროგრამის შესრულების დასაწყისში.

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

- TL, TR, BL, BR: მიღებული ბლოკის ზედა-მარცხენა, ზედა-მარჯვენა, ქვედა-მარცხენა და ქვედა-მარჯვენა ფილების ფერები, შესაბამისად, ისე როგორც სურათზე ნაჩვენები. თითოეული მნიშვნელობა არის 0(თეთრი) ან 1(შავი).
- ეს პროცედურა გამოიძახება ზუსტად $N \cdot M$-ჯერ თითოეულ ტესტში, `init`-ის გამოძახების შემდეგ.



ამ პროცედურამ უნდა დააბრუნოს მთელი რიცხვების წყვილი (i, j), სადაც i არის იმ უჯრის სტრიქონის კოორდინატი, ხოლო j — სვეტის კოორდინატი, სადაც ამ ბლოკის ზედა მარცხენა ფილა უნდა განთავსდეს. i და j უნდა იყოს ლუწი რიცხვები, ხოლო ბლოკით დაფარული 2×2 რეგიონი არ უნდა იკვეთებოდეს რომელიმე აქამდე განთავსებულ ბლოკთან.

თუ `receive_block`-მა ოდესმე დააბრუნა წყვილი, რომელიც არ აკმაყოფილებს ამ მოთხოვნებს, ან თუ ბლოკის განთავსების შემდეგ რომელიმე 2×2 კვადრატი სრულად დაიფარება შავი ფილებით, გრეიდერი დაუყოვნებლივ შეწყვეტს თქვენი პროგრამის მუშაობას და ვერდიქტი იქნება `Output isn't correct`.

გრეიდერი **არაა ადაპტიური**. ეს ნიშნავს რომ ბლოკების მიმდვერობა, რომელსაც ბარჩინი აძლევს ქაროსს `init`-ის გამოძახებამდეა დაფიქსირებული.

შეზღუდვები

თითოეული ბლოკისთვის, S იყოს მისი 4 ფილიდან შავად გაფერადებული ფილების რაოდენობა. ანუ $S = TL + TR + BL + BR$.

- $1 \leq N, M \leq 100$
- $0 \leq S \leq 3$ ყოველი ბლოკისთვის.

ქვეამოცანები

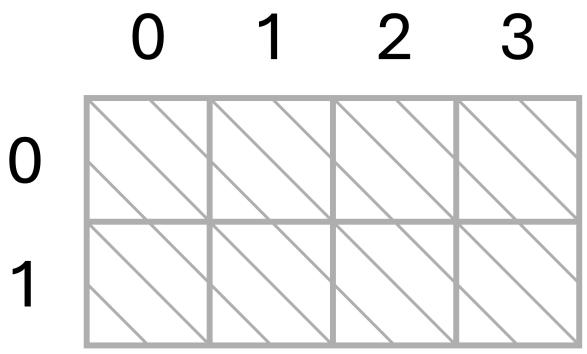
ქვეამოცანა	ქულა	დამატებითი შეზღუდვები
1	6	$S = 1$ ყველა ბლოკისთვის და $N = 2$.
2	16	$S = 3$ ყველა ბლოკისთვის და $N = M$, N ლუწია, და ოთხივე შესაძლო გაფერადება გვხვდება ზუსტად $\frac{N^2}{4}$-ჯერ.
3	10	$S = 1$ ყველა ბლოკისთვის.
4	29	$S \leq 2$ ყველა ბლოკისთვის.
5	39	დამატებითი შეზღუდვების გარეშე.

მაგალითი

განვიხილოთ თამაში $N = 1$ and $M = 2$, მაშასადამე დაფას აქვს 2 სტრიქონი და 4 სვეტი. გრეიდერი თავდაპირველად იძახებს :

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

თავდაპირველად, ყველა უჯრა ცარიელია, დაფა შემდეგნაირად გამოიყურება:

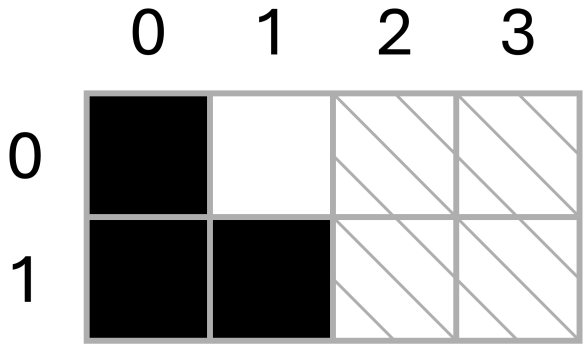


$N \cdot M = 2$ ცალი ბლოკია განსალაგებელი. დავუშვათ ბარჩინი იძლევა ბლოკს 3 შავი ფილით და 1 თეთრი ფილით ზედა მარჯვენა კუთხეში. გრეიდერი იძახებს:

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

ქაროსმა გადაწყვიტა ეს ბლოკი დადოს დაფის მარცხენა მხარეს და აბრუნებს $(0, 0)$.

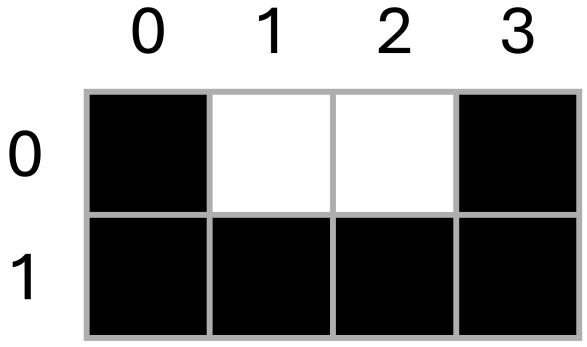
დაფა ამჟამად ასე გამოიყურება:



ბარჩინი ახლა იძლევა ბლოკს სამი შავი ფილით, რომელსაც თეთრი ფილა აქვს ზედა-მარცხენა კუთხეში

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

ერთადერთი დარჩენილი უჯრა ლუწი სტრიქონით და ლუწი სვეტით, რომლიც შეიძლება იყოს დადებული 2×2 ბლოკის ზედა-მარცხენა კუთხე არის $(0, 2)$, ამიტომ ქაროსი აბრუნებს $(0, 2)$. დაფა ასე ივსება:



არცერთი 2×2 კვადრატი არაა ბოლომდე შავი, ანუ ქაროსმა წარმატებით განათავსა ყველა ბლოკი ისე რომ ბარჩინს არ მოუგია. შესაბამისად, ქაროსი იგებს.

სანიმუშო გრეიდერი

შეტანის ფორმატი:

```
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]
```

გამოტანის ფორმატი:

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

აქ, $R[k]$ and $C[k]$ არის მთელი რიცხვების წყვილი, რომელიც დააბრუნა `receive_block`-ის k-ე გამოძახებამ .