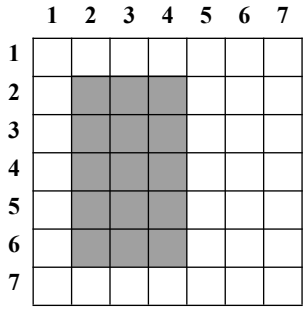
XOR

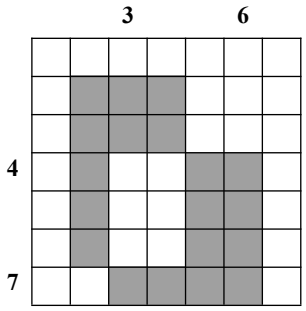
ЗАДАЧА

Предстои да напишете програма за управление на мобилен телефон с черно-бял екран. Номерацията на *x*-координатите на екранните пиксели е отляво надясно, а на *y*-координатите – отгоре надолу, както е показано на фигурите. За програмата са необходими различни изображения, които не са непременно с еднакви размери. Вместо да ги зареждате всеки път, решавате да ги създавате с използване на графичната библиотека на мобилния телефон. Предполага се, че в началото на изчертаването на изображение, всички пиксели на екрана са бели. Единствената графична операция в библиотеката на мобилния телефон е `XOR(L,R,T,B)`, която променя цвета на всички пиксели от правоъгълника с най-горна най-лява координата (`L`,`T`) и най-долна най-дясна координата (`R`,`B`), където `L` означава ляво, `T` – горе, `R` – дясно, а `B` – долу. Забележете, че в някои други графични библиотеки редът на параметрите е различен.

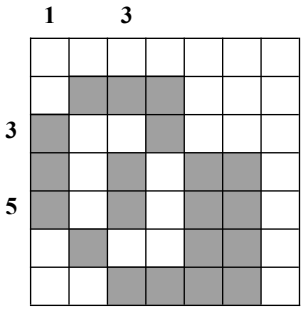
Като пример разгледайте изображението от Фигура 3. Прилагайки операцията `XOR(2,4,2,6)` към чисто бял екран, се получава изображението от Фигура 1. Прилагайки опрацията `XOR(3,6,4,7)` към изображението от Фигура 1, се получава изображението от Фигура 2, а прилагайки операцията `XOR(1,3,3,5)` към изображението от Фигура 2, се получава изображението от Фигура 3.



Фигура 1



Фигура 2



Фигура 3

По зададено множество от черно-бели изображения, задачата ви е да получитете всяко изображение, започвайки от начален бял екран и използвайки колкото можете по-малко XOR операции. Ще ви бъдат предоставени входни файлове с описанията на изображения, а вие трябва да изпратите файлове с намерените от вас параметри на XOR операции, а не програма, която да ги генерира.

ВХОД

Входните данни за 10 тестови примера ще бъдат във файловете с имена от `xor1.in` до `xor10.in`. Всеки входен файл е оформен както следва. Първият ред съдържа цялото число N, $5 \le N \le 2000$, означаващо, че екранът е с N реда и N стълба. Останалите N реда описват по един от редовете на изображението отгоре надолу. Всеки ред съдържа по N числа – стойностите на пикселите в реда

отляво надясно. Всяко от тези числа е или 0 или 1, като 0 означава бял, а 1 – черен пиксел.

ИЗХОД

Трябва да изпратите десет изходни файла, съответни на зададените входни файлове.

Първият ред трябва да съдържа текста

```
#FILE xor I
```

Където `I` е номерът на съответния входен файл. Вторият ред съдържа едно цяло число *K* – броят на XOR операциите описани по-надолу във файла. Следващите *K* реда съдържат параметрите на XOR операциите, започвайки с първата операция, която трябва да се изпълни и завършвайки с последната. Всеки от редовете съдържа по четири цели числа, разделени с интервал – параметрите `L`, `R`, `T`, `B` на съответната операция в указания ред.

ПРИМЕРЕН ВХОД И ИЗХОД

Пример: xor0.in xor0.out

```
7
0 0 0 0 0 0 0
0 1 1 1 0 0 0
1 0 0 1 0 0 0
1 0 1 0 1 1 0
1 0 1 0 1 1 0
0 1 0 0 1 1 0
0 0 1 1 1 1 0
```

```
#FILE xor 0
3
2 4 2 6
3 6 4 7
1 3 3 5
```

ОЦЕНЯВАНЕ

Ако

- XOR операциите в изпратения изходен файл не създават исканото изображение, или
- в изходния файл броят на XOR операциите не е *K*, или
- в изходния файл $K > 40000$, или
- в изходния файл се среща XOR операция такава че L>R или T>B, или
- в изходния файл се среща XOR операция с не положителни параметри, или
- в изходния файл се среща XOR операция с параметър надхвърлящ *N*,

тогава ви се присъждат нула точки. В противен случай получавате
$1 + 9 \times$ CallsInBestAnswerOfAllContestants / CallsInYourAnswer точки.

За всеки тестов пример получените по формулата точки се закръгляват до първата цифра след десетичната точка. Сумата от точки за всички тестове се закръглвая до най-близкото цяло число.

Например, нека сте изпратили решение със 121 XOR операции. В случай, че това е най-доброто от всички изпратени решения, ще получите всичките 10 точки за теста. Ако най-доброто изпратено от участник решение е с 98 XOR-операции, тогава ще получите $1+9\times98/121$(=8.289…), което се закръглява до 8.3.