

За разлика от фирмата [Skyscanner](#), която предоставя търсачка за изгодни самолетни билети, хотели и коли под наем, фирмата Spyscanner се занимаваше с нещо съвсем друго – монтираше малки камери отдолу по корпусите на самолетите и правеше снимки отвисоко докато те летяха насам-натам.

Оказа се, че камерите имат дефект и вместо за всеки пиксел да задават неговите **RGB стойности** - число между 0 и 255 (включително) за червената, зелената и синята компонента - те връщаха осреднения цвят на всички пиксели от района, който е бил заснет. Така, например, ако заснетият район е бил българското знаме (1/3 бяло (255, 255, 255), 1/3 зелено (0, 255, 0) и 1/3 червено (255, 0, 0)), резултатът е един единствен цвят с трите си компоненти: (170, 170, 85). Получихме него, тъй като  $(255 + 0 + 255) / 3 = 170$ ,  $(255 + 255 + 0) / 3 = 170$  и  $(255 + 0 + 0) / 3 = 85$ .

Програмистите от Spyscanner не са се отчаяли! Те вярват, че ползвайки многократно камерите, могат да постигнат добро приближение на оригиналната снимка, въпреки дефекта. Служителите са решили да снимат правоъгълни под-региони от района, който искат да изучат, като така получават информация за произволни негови части.

Вие отскоро сте в компанията и искате да блеснете, написвайки изкуствен интелект, който прави това!

## Задача

Дадени са ви целите числа **N** и **M** – съответно броят пиксели по редове и колони на избрана от нас картинка. Индексирането по редове е от горе надолу, а по колони – от ляво надясно. Така най-горният ляв пиксел е с координати (0, 0), а най-долният десен – с координати (**N**-1, **M**-1).

Допълнително ви е дадено цялото число **K** - колко въпроси можете да зададете. Всеки от въпросите се задава чрез горния ляв ъгъл (**R<sub>1</sub>**, **C<sub>1</sub>**) и долния десен ъгъл (**R<sub>2</sub>**, **C<sub>2</sub>**) на под-правоъгълник от картинката, за който се интересувате. Правоъгълниците, за които питате, са със страни, успоредни на страните на картинката. Те могат да бъдат с произволен размер (включително цялата картинка или пък един единствен пиксел от нея). Разрешено е да се припокриват частично или дори изцяло.

Отговорът на всеки въпрос ще е осредненият цвят за избрания правоъгълник (средното аритметично на червените компоненти на всички пиксели в него; средното аритметично на зелените компоненти на всички пиксели в него; и средното аритметично на сините компоненти на всички пиксели в него). При осредняването се ползва целочислено деление – така отговорът на всеки въпрос отново е стандартен RGB цвят.

Забележете, че задаването на въпроси е интерактивно – тоест може да зададете въпрос, да прочетете отговора и чак след това да зададете следващия въпрос!

От вас се иска да изведете картинка с **N** реда и **M** колони – вашето приближение на оригиналната такава.

## Вход и Изход

При пускането на програмата ви, на стандартния вход (stdin) ще са зададени трите цели числа **N**, **M**, и **K**. За да зададете въпрос изпечатайте на стандартния изход (stdout) ред с четири цели числа **R<sub>1</sub>** **C<sub>1</sub>** **R<sub>2</sub>** **C<sub>2</sub>**, разделени с интервали. Отговорът се чете от стандартния вход. Той ще е представен като ред с три цели числа между 0 и 255 (включително), разделени с интервали – средното аритметично на всяка от трите компоненти за правоъгълника, за който питате.

За да се избегне забавяне на програмата в следствие на буфериране на изхода, след изпечатването на всеки въпрос flush-вайте изхода:

- C: `fflush(stdout);`
- C++: `cout << flush;`
- Java: `System.out.flush();`
- Python: `stdout.flush()`

За да изведете крайния резултат, на стандартния изход първо изпечатайте един ред с думата "Ready", а след това **N** реда, всеки съдържащ по **M** \* 3 цели числа между 0 и 255 (включително) – **M**-те пиксела за съответния ред, всеки с трите си компоненти.

## Ограничения

- $1 \leq N, M \leq 512$
- $0 \leq R_1 \leq R_2 \leq N - 1$
- $0 \leq C_1 \leq C_2 \leq M - 1$
- В 30% от тестовете  $1 < K \leq 1,000$
- В 30% от тестовете  $1,000 < K \leq 5,000$
- В 40% от тестовете  $5,000 < K \leq 10,000$

Примерен Вход	Примерен Изход
5 3 4 170 171 91 171 170 91 7 250 8 25 255 0	0 0 4 2 1 0 3 2 0 1 4 1 2 1 2 1 Ready 250 0 135 7 250 8 250 0 135 255 0 135 7 250 8 255 0 135 255 0 135 25 255 0 255 0 135 255 0 135 7 250 8 255 0 135 250 0 135 7 250 8 250 0 135

Картинката, която искаме да намерим, има 5 реда и 3 колони.

Имаме право на 4 питаня. В първото искаме да разберем каква е средната стойност на цветовете в целия регион (от (0, 0) до (4, 2)). Този цвят е (170, 171, 91). На второто питане взимаме по-малък правоъгълник, отново обхващащ и трите колони, получавайки (171, 170, 91). В третото питане искаме да видим цвета на средната колона, получавайки (7, 250, 8). В последното питане искаме цвета на специфичен пиксел – именно този в центъра на картинката. Неговият (точен) цвят е (25, 255, 0).

Забележете, че изведеният след това отговор описва цялата картинка пиксел по пиксел. Как точно се ползва информацията от въпросите до сега зависи изцяло от вас!

### Примерна картинка

Примерна картинка в .bmp формат можете да изтеглите от [тук](#), а представена в текстов формат от [тук](#) (първият ред са числата **N**, **M** и **K**, а на следващите **N** реда има по **M** \* 3 числа, описващи картинката).

### Оценяване

Ако използвате повече от **K** въпроса, някой от въпросите е невалиден или не спазва зададения формат, изведеният отговор е невалиден или не спазва зададения формат, използвате повече от разрешеното време, използвате повече от разрешената памет, или по някаква причина решението ви не завърши успешно (crash-не), то ще получите 0 точки за този тест.

В случай, че всичко е наред, точките за съответния тест ще бъдат формирани по следния начин. Нека дефинираме разстояние между два пиксела като сумата от разликите на квадрат на трите им компоненти. Например, ако единият пиксел е бил с цвят (42, 13, 202), а другият - (40, 50, 180), то разстоянието между тях е  $(42 - 40)^2 + (13 - 50)^2 + (202 - 180)^2 = 4 + 1369 + 484 = 1857$ . Нека сумата от разстоянията за всички **N** \* **M** пиксела между оригиналната картинка и вашия отговор е числото **Your**, а най-малката такава сума на някой от участниците е **Best**. Вие ще получите  $(Best + 1) / (Your + 1)$  от точките за този тест. С други думи, целта е да се минимизира сумарното разстояние между вашия отговор и истинската картинка.