

Ели и Крис играят на следната игра. Крис рисува с молив на бял лист хартия някаква картинка. След това Ели се опитва да я реконструира, питайки "Насочената отсечката от (X_1, Y_1) до (X_2, Y_2) пресича ли рисунката, и ако да – коя е първата точка, в която го прави?". След определен брой въпроси Ели трябва да нарисува нейна картинка, базирана на информацията, която е получила. Разбира се, целта на Ели е нейната картинка да се доближава възможно най-много до оригиналната.

Задача

Вие искате да впечатлите Ели, като напишете програма, която прави нейните действия и постига възможно най-добра реконструкция на оригиналната картинка за дискретна версия на горната игра.

Рисунката на Крис ще е черно-бяла (бинарна) картинка с размери 512 на 512 пиксела. Вместо с координати (X, Y) , за удобство ще индексирате пикселите на картинката с ред и колона (R, C) , където най-горният ляв пиксел е с координати $(0, 0)$.

Можем да зададем не повече от K въпроса, където K е естествено число, зададено на стандартния вход при стартиране на програмата ни. Въпросите задаваме изпечатвайки на стандартния изход $R_1 C_1 R_2 C_2$ - четири цели числа в интервала $[0, 511]$, задаващи реда и колона, от които започва насочената отсечка, следвани от ред и колона, в които свършва. За всеки въпрос на стандартния вход получаваме две цели числа – двойката $-1 -1$, ако отсечката от (R_1, C_1) до (R_2, C_2) не съдържа черен пиксел, или двойката $R C$, задаваща първия черен пиксел по насочената отсечка. За дискретизиране на насочената отсечка ще бъде ползвана [модификация на алгоритъма на Брезенхам](#).

Забележете, че когато задаваме въпрос, можем да ползваме информацията, която вече сме получили от предходните такива.

Когато сме готови с нашата реконструкция на картинката трябва да изпечатаме на отделен ред думата "Ready", а след това 512 реда всеки с по 512 символа '0' и '1', задаващи, съответно, бял и черен пиксел. Не трябва да печатаме интервали между нулите и единиците на един ред.

За да се избегне забавяне на програмата в следствие на буферизиране на изхода, след изпечатването на всеки въпрос, а както и след изпечатването на финалния отговор, flush-вайте изхода:

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

Визуализатор

За ваше удобство задачата има вграден визуализатор. След като предадете решение, всеки "зелен" резултат може да бъде цъкнат, с което се отваря визуализаторът за съответната картинка.

Ограничения

- $1,000 \leq K \leq 5,000$

Примерен Вход	Примерен Изход
1337 42 42 -1 -1 42 17	0 0 511 511 0 0 0 511 42 0 42 42 Ready 000000000000... 0000011010001... . . .

Имаме право да зададем до 1337 въпроса. Първо питаме за отсечката $(0, 0) \rightarrow (511, 511)$, за която научаваме, че първият черен пиксел по нея се намира в $(42, 42)$. След това задаваме въпрос за хоризонталната линия $(0, 0) \rightarrow (0, 511)$, за която научаваме, че няма черен пиксел. Последно питаме за $(42, 0) \rightarrow (42, 42)$, като научаваме, че има черен пиксел в $(42, 17)$. Забележете, че макар и да има друг черен пиксел в $(42, 42)$, отговорите на въпросите задават първия срещнат такъв. Накрая сме решили да спрем с въпросите и да изпечатаме финалния ни отговор.

Примерният изход е съкратен по очевидни причини.

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 $ourScore$ бъде броят познати от нас пиксели за тази картинка, а $bestScore$ бъде най-големият такъв брой измежду всички участници. Точките ни за този тест ще бъдат равни на $5 * (ourScore / bestScore)^3$ - релативно оценяване спрямо най-доброто решение на куб.

Временно и финално класиране

По време на разработването на решения (в интервала на състезанието) ще бъдат дадени 20 картинки, на базата на които ще бъде изготвяно временно класиране, достъпно за всички участници. След края на състезанието ще бъдат качени 20 нови картинки, на базата на които ще се изготви финалното класиране. Забележете, че първите 20 няма да допринесат за точките на участниците във финалното класиране, но ще са донякъде подобни (със сходни характеристики) на

финалните.

Update 1: Финалното класиране можете да намерите [ето тук](#).

Update 2: Състезанието приключи. Качих всички 40 теста на задачата и предаването на решения е отново отворено.