

Есенно почистване

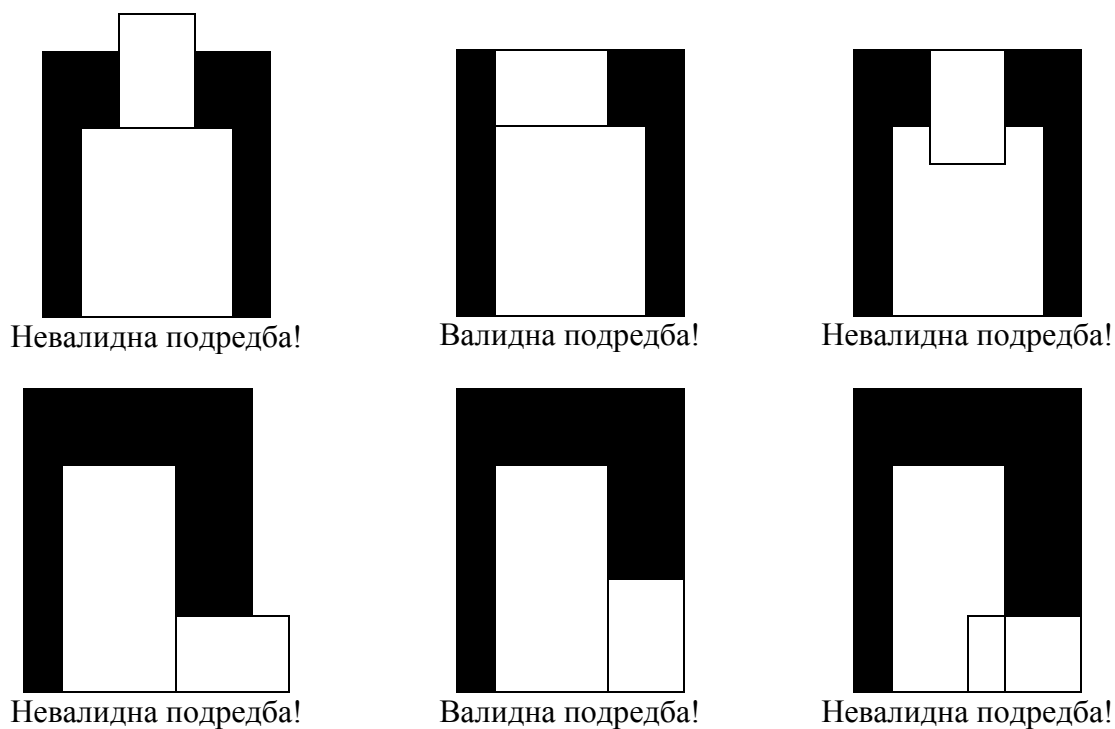
С помощта на Бухал, Мечо Пух се прибра успешно у дома. След моментната уплаха от изгубването, той отново се върна към основното си занимание – мислене колко е хубаво да имаш много мед. Въпреки лошите спомени от горската поляна у него остана светлият образ на многото бурканчета, които все още седяха на нея. Воден от максимата си „Колкото повече, толкова повече!!!”, Мечо Пух твърдо реши да има поне едно гърненце от всеки вид мед, предлаган на пазара. Според него тазгодишната реколта притежава уникален вкус, получен от смесването на местния цветен прашец и исландския такъв, донесен от изригването на вулкана Ейяфятлайокул, и ще е голяма загуба за неговата колекция, ако не ги притежава. За жалост, рафтовете в мазето на Пух са пълни с бурканчета и трябва да се освободи място за новите, но той няма намерение да изхвърля никаква храна. Затова Мечо Пух има добре изградена стратегия. Той ще прибере всички буркани от рафтовете в кашони, които ще складира на тавана си, и така ще освободи място. Разбира се, има завидно количество мед в мазето и при невнимателно подреждане ще се използват прекалено много кашони, които от своя страна струват пари. За Мечо Пух няма по-добра инвестиция от бурканчета с мед, затова иска да се използва минимален брой кашони при пренасянето и да останат повече пари за покупки. Пух не се слави с прозорливост и не би се справил бързо с тази задача, а колкото по-рано отиде на пазара, толкова шансът да има повече различни видове мед е по-голям. Той разчита на вас да напишете програма **cleaning**, която намира ефикасен начин за разпределянето на различните бурканчета в отделните кашони.

Във всеки кашон бурканчетата ще се редят в един ред и затова, всяко бурканче с мед е представено с основата си – правоъгълник със страни a_i и b_i . Аналогично, всеки от кашоните е правоъгълник с дължина A и ширина B .

На първия ред на входния файл **cleaning.in** е зададено цяло положително число N – броят на бурканчетата по рафтовете в мазето на Мечо Пух. На втория ред са зададени двете цели положителни числа A и B ($A \geq B$). На всеки от следващите N реда са зададени по две цели положителни числа a_i и b_i ($a_i \geq b_i$) – размерите на i -тото бурканче.

Исходният файл **cleaning.out** трябва да съдържа $N+1$ реда. На първия ред стои едно цяло число – броят на използваните кашони. На $i+1$ -я ред от изхода трябва да бъде изведена информацията за i -тото бурканче с мед от входния файл: номера на кашона, в който ще бъде поставено съответното бурканче, координатите на долния му ляв ъгъл в кашона и страната, на която лежи бурканчето – буквата ‘**a**’, ако бурканчето е поставено с по-дългата си страна към страната на кашона с дължина A , или ‘**b**’, ако е поставено с по-късата си страна. Начало на координатната система на всеки кашон е долният му ляв ъгъл. Естествено, две бурканчета може да се допират, но не може едното да застъпва другото с части от вътрешността си. Всяко от бурканчетата може да се допира до страна на кашона, но не може да излиза извън него. Не е необходимо бурканчетата да се редят “плътно” – празни пространства са допустими във всяка посока.

Пояснение:



cleaning.in	cleaning.out
7	2
8 7	1 0 0 b
3 3	1 3 0 b
5 4	2 4 3 a
2 2	1 7 0 b
6 1	1 0 6 a
7 1	2 1 0 b
4 2	2 3 0 a
4 3	

<p>Кашон 1</p>	<p>Кашон 2</p>
----------------	----------------

Ограничения:

$1 < N < 5000$ и $3 < B \leq A < 256$

Гарантирано е, че всяко бурканче може да се побере в кашон с размери **A** и **B**.

Оценяване:

При некоректен изход получавате 0 точки за съответния тест. В противен случай, намерения от програмата Ви брой на необходимите кашони ще се сравни с намерени от останалите участници и ще получите точки пропорционално на качеството на Вашето решение.