С Ваша помощ Иванчо успя да закърпи финансовото си положение и е готов да се завърне към науката! Сега задачата, с която се е захванал, е да реши система линейни уравнения.

Система линейни уравнения представлява следното:

При зададени *ai,j*, *1 ≤ i ≤ m*, *1 ≤ j ≤ n* и bi, *1 ≤ i ≤ m* се търсят стойностите *xj*, *1 ≤ j ≤ n*.

След известно време размисли Иванчо осъзна, че би било твърде амбициозно (а понякога и невъзможно) да се търси решение на произволна система линейни уравнения. Затова той си постави по-реалистична цел – да намери *n*-торка (*x1, … , xn*), която да бъде възможно най-близка до пълно решение.

За тази цел за всяко уравнение се задават две стойности *exacti* и *aproxi*, които определят какъв резултат получава дадено решение. Първата стойност е бонусът, който решението получава, ако е вярно решено i-тото уравнение. Втората стойност определя бонус по следната формула (пресмята се отделно за всяко уравнение i):

Резултатът се образува като сбор на двата бонуса.

Ясно е, че всеки момент на Иванчо ще му стане скучно и отново ще остави задачата на Вас, но поне Ви дава право на избор – дава Ви *s* на брой *m*-торки (*b1*, …, *bm*) и Вие избирате с коя *m*-торка да образувате системата, която ще трябва да решите.

Напишете програма, която определя за коя *m*-торка ще решавате системата и намира *n*-торка от *x*-ове, получаваща възможно най-голям резултат за избраните *b*-та.

**Вход**

От първия ред на файла equations.in се въвеждат две цели положителни числа *m* и *n* – съответно броят на уравненията в системата и броят на неизвестните. Следват *m* реда с по *n+2* цели чиста на ред – коефициентите *ai,j*, *exacti* и *aproxi*. От следващия ред се прочита броя *s* на *m*-торките (*b1*, …, *bm*), а следващите *s* реда съдържат самите *m*-торки, т.е. *m* на брой цели числа.

**Изход**

На първия ред на изходния файл equations.out трябва да се изведе номерът на избраната *m*-торка (индексирането започва от 1). На втория ред да се изведат *n* числа – *x1*, … , *xn*, отделени с интервали. Тези числа трябва да бъдат цели и абсолютната им стойност да не надвишава 106.

**Оценяване**

Ще получите 0 точки, ако изходът Ви не удовлетворява поставените условия и ограничения. В противен случай ще получите процента от точките, предвидени за съответния тест. Дефинираме yourScore като резултатът, който е получила Вашата програма на съответния тест, а maxScore като най-големия резултат, който е получила някоя от програмите на участниците на този тест.

**Ограничения**

1 ≤ m ≤ 2000

1 ≤ *n* ≤ 1000

1 ≤ s ≤ 20

0 ≤ aij ≤ 105

103 ≤ exacti, aproxi ≤ 106

0 ≤ bi ≤ 109 за всяка *m*-торка.

Забележка: Стойностите на aij, exacti, aproxi и bi са произволно генерирани, така че да удовлетворяват ограниченията.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Брой тестове** | 25% | 25% | 25% | 25% |
| ***m*** | 800 | 1000 | 1500 | 2000 |
| **n** | 1000 | 900 | 1000 | 1000 |
| **s** | 1 | 10 | 15 | 20 |

**Ограничение по време: 5 сек**

**Ограничение по памет: 256 MB**

**Примерен тест**

|  |  |
| --- | --- |
| **Вход (equations.in)** | **Изход (equations.out)** |
| 5 4  6456 14482 80443 91712 856827 404250  46603 85084 90886 43276 359810 220421  73805 76890 42862 62163 621170 795841  6324 15062 37884 17771 890601 261041  53674 13346 95893 53116 150825 207655  1  175231511 46012106 293888266 414081012 725440535 | 1  713 984 870 942 |

**Обяснение на тестовия пример**

Примерният изход получава резултат 20212 (стойността се разглежда като цяло число). Не се гарантира, че това е оптимален резултат.