With your aid Ivancho managed to restore his finances so he is ready to get back to science! Now he is focused on the problem of solving a system of linear equations.

A system of linear equations is the following:

Given *ai,j*, *1 ≤ i ≤ m*, *1 ≤ j ≤ n* и bi, *1 ≤ i ≤ m* you are asked to find *xj*, *1 ≤ j ≤ n*.

After some intense thinking Ivancho realized that it would be too ambitious (and sometimes even impossible) to search for a solution of a random system of linear equations. That is why he set a more realistic goal – to find an *n*-tuple (*x1, … , xn*), which is as close as possible to a complete solution.

For this purpose you are given two values *exacti* and *aproxi* for each equation, which determine the result of a solution. The first value is the bonus for successfully solving the i-th equation. The second value determines the bonus by the following formula (computed separately for each equation i):

The result is formed as a sum of the two bonuses.

It is obvious that any second Ivancho will get bored and will leave the problem to you, but at least he gives you the right to choose – he gives you *s* *m*-tuples (*b1*, …, *bm*) and you get to choose which *m*-tuple to use to form the system which will need to be solved.

Write a program that determines the *m*-tuple for which you are going to solve the system and then finds an *n*-tuple of *x*-es which receives as big of a result as possible for the chosen *b*’s.

**Input**

The first line of the input file equations.in contains two positive integers *m* and *n* – corresponding to the number of equations in the system and the number of variables. It is followed by *m* lines with *n+2* integers per line – the coefficients *ai,j*, *exacti* and *aproxi*. The new line contains the number *s* of *m*-tuples (*b1*, …, *bm*), and the next *s* lines contain these *m*-tuples, i.e. *m* integes.

**Output**

On the first line of the output file equations.out write the number of the chosen *m*-tuple (indexed from 1). On the second line output *n* numbers – *x1*, … , *xn*, separated by spaces. They must be integers with their absolute value not exceeding 106.

**Scoring**

You will receive 0 points if your output does not satisfy the stated conditions and constraints. Otherwise, you will receive percent of the points for each test. We define yourScore as the result of your program for the current test and maxScore as the highest result received by some of the contestants’ programs for this test.

**Constraints**

1 ≤ m ≤ 2000

1 ≤ *n* ≤ 1000

1 ≤ s ≤ 20

0 ≤ aij ≤ 105

103 ≤ exacti, aproxi ≤ 106

0 ≤ bi ≤ 109 for each *m*-tuple.

Notice: The values of aij, exacti, aproxi and bi are randomly generated so that they satisfy the constraints.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Number of tests** | 25% | 25% | 25% | 25% |
| ***m*** | 800 | 1000 | 1500 | 2000 |
| **n** | 1000 | 900 | 1000 | 1000 |
| **s** | 1 | 10 | 15 | 20 |

**Time limit: 5 sec**

**Memory limit: 256 MB**

**Example**

|  |  |
| --- | --- |
| **Input (equations.in)** | **Output (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 |

**Explanation**

The sample output receives result 20212 (the value is casted to integer). It is not guaranteed to be the optimal result.