French-Australian Regional Informatics Olympiad

12th March, 2010

Duration: 4 hours

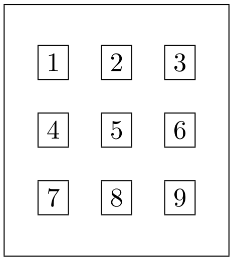
4 questions

All questions should be attempted

Problem 1
The FARIO Incident

Time and Memory Limits: 1 second, 16 MB

You are a superspy. Your job requires you to travel around the world, visiting exotic locales and driving expensive cars. Your latest mission is to break into the headquarters of the evil F.A.R.I.O. organisation and steal important documents. Getting in is easy for a spy like you, but the documents are in a safe which can only be opened by entering the correct code into the keypad.



The keypad is a 3×3 grid numbered from 1 to 9, as shown above. Your sources tell you that *every two consecutive digits in the code will always be adjacent keys on the keypad.* For example, the digit 1 will only be followed by a 2 or 4, the digit 2 will only be followed by a 1, 3 or 5, and so on. So 3252 and 12369 are valid codes, but 1234 is not (3 is not adjacent to 4 on the keypad) and 55 is not (5 is not adjacent to 5 on the keypad).

You also know how many digits there are in the code, and the first digit. Given this information, how many possible secret codes are there?

Input

Your program must read from standard input. The input will consist of two space-separated integers, N and D. N is the number of digits in the code ($1 \leq N \leq 1\,000\,000$), and D is the first digit ($1 \leq D \leq 9$).

For 30% of the available marks, $N \leq 10$.

Output

Your program must write to standard output. The output should consist of a single integer: the number of possible codes, *modulo* 1 000 003.

Sample Input 1

```
3 4
```

Sample Output 1

```
8
```

Sample Input 2

```
42 8
```

Sample Output 2

```
128254
```

Explanation

In sample input 1, a code of length 3 and starting from the digit 4, the possible codes are 412, 414, 452, 454, 456, 458, 474 and 478.

Scoring

The score for each input scenario will be 100% if the correct answer is written to the output file, and 0% otherwise.

Problem 2
ABC...

Time and Memory Limits: 5 seconds, 32 MB

Despite your best efforts, your writing is riddled with typos. You are constantly trying to improve, but it is hard work. Luckily, you have just come up with a brilliant solution – you will write a computer program to correct misspelt words!

Your 'misspell checker' must be able to take a dictionary of valid words and a message, and correct all the typos. For the purposes of this problem, a *typo* is a word not in the dictionary which, by replacing a single letter with a different letter, becomes a valid word.

Input

The first line of input will contain a single integer N, indicating the number of words in the dictionary ($1 \leq N \leq 100\,000$). Each of the next N lines will contain a single distinct dictionary word. The next line of input will contain a single integer M, indicating the number of words in the message ($1 \leq M \leq 10\,000$). Each of the next M lines will contain a single word to be checked. All words will consist of between 1 and 20 lower-case letters, inclusive.

For 30% of the available marks, $N, M \leq 1000$. For 60% of the available marks, $N \leq 30\,000$.

Output

For each of the M message words in the input, your program should output a single line:

- If the input word was correctly spelt (i.e., it is in the dictionary), output it as it is;
- If the input word can be corrected by replacing a single character, output the corrected version. If there are multiple possibilities, output any one of them;
- If the input word cannot be corrected by replacing a single character, output the single character '?'.

Sample Input

```
9
i
apple
far
mat
job
for
then
may
apply
6
may
i
apple
fur
the
jpb
```

Sample Output

```
may
i
apple
for
?
job
```

Scoring

The score for each input scenario will be 100% if a correct answer is written to the output file, and 0% otherwise.

Problem 3
Dating Site IT

Time and Memory Limits: 3 seconds, 64 MB

You run an online dating site with many users. You keep detailed statistics for each user, including the number of times they visit your site this month. You would like to know how many visits your site gets in total this month. Unfortunately this is not as easy as it sounds – some of your users are *spammers*, who use your site to advertise their own products. You do not want to count the spammers' visits towards your total.

When a user registers with the site, they must supply many personal details including their age, height and weight. Spammers lie about these details, but they are very predictable and you can use their fake answers to spot them. For example, you might know that "any user whose age is between 18 and 30, whose height is between 80cm and 90cm, *and* whose weight is between 40kg and 50kg is a spammer".

Your task is to write a computer program that takes a number of statements like the above and a list of user statistics, and outputs the total number of visits your site has received this month.

Input

Your program must read from standard input. In the description below, a_i, h_i, w_i, v_i represent a user's age, height, weight and number of visits this month, respectively.

The first line will consist of two space-separated integers, S and U ($1 \leq S, U \leq 1\,000\,000$). Each of the following S lines will be of the form 'a_{min} a_{max} h_{min} h_{max} w_{min} w_{max}', representing the statement "any user that satisfies $a_{min} \leq a_X \leq a_{max}$ *and* $h_{min} \leq h_X \leq h_{max}$ *and* $w_{min} \leq w_X \leq w_{max}$ is a spammer". (Of course, $a_{min} \leq a_{max}$ and so on). Following this will be U lines describing a single user, in the form 'a_i h_i w_i v_i'.

All ages, heights and weights are integers between 0 and 99 inclusive. Each user will make between 1 and 1000 visits, inclusive.

See the *Scoring* section for a breakdown of the marks awarded for different test cases.

Output

Your program must write to standard output. The output should consist of a single integer: the number of visits made to your site this month by users who are *not* spammers.

Sample Input

```
2 3
18 24 30 60 80 85
21 30 40 50 0 99
18 30 80 3
25 35 86 5
23 45 82 7
```

Sample Output

```
5
```

Scoring

The score for each input scenario will be 100% if the correct answer is written to the output file, and 0% otherwise.

The following table lists the sizes of the inputs used in the test data. Here, $N - 1$ is the maximum age, height and weight specified in the input.

N	S	U	Percentage of marks
20	10	1000	10%
20	100	8000	10%
10	10000	100000	10%
30	10000	1000000	10%
50	100000	100000	20%
100	1000000	500000	20%
100	1000000	1000000	20%

Problem 4
Human LCD

Time and Memory Limits: 10 seconds, 64 MB

You are working hard to make informatics competitions more entertaining to the public, and are organizing this year's olympiad in a big indoor stadium, with a big audience. While all the contestants will try hard to concentrate on solving problems, cheerleaders will cheer as scores appear on giant screens in real time, the olympiad's mascot will throw small gifts to the crowd and guides will perform as a giant human LCD screen.

All this costs more than expected and you had to find an extra sponsor at the last minute. A fast food chain has agreed to pay for the extra costs on the condition that their logo be added to the end of the human LCD routine.

At the end of the existing routine, the guides will be arranged as a nice 2-dimentional grid. Each cell will contain a guide wearing a coloured t-shirt, forming one pixel of the human LCD screen. You are given a description of this grid and the colour of the guide contained in each cell. You are also given a description of the colour that each cell should display to represent the sponsor's logo perfectly.

As you don't have much time to train the guides for this extra move, you want to make it simple: each guide will either not move, or swap their position with one of the guides in the 4 adjacent cells. All swaps happen simultaneously, and no guide may be involved in more than one swap. Note that unlike real human LCDs, the guides cannot change their colour.

Your goal is to find a collection of such swaps, so that the resulting image formed on the grid is as close as possible to the sponsor's logo.

Input

The first line of input will contain two space-separated integers R, C: the number of rows and columns of the grid ($1 \leq R, C \leq 100$).

The following R lines will describe the initial grid as it is shown to the public at the end of the original routine. Each line describes a row of the grid and contains C space-separated integers, the colours of the guides in the cells of that row. All colours are integers between 0 and 9, inclusive.

The following R lines will describe the grid that you are trying to obtain, representing the sponsor's logo. Each line describes a row and contains C space-separated integers, the colours the cells should show to form the sponsor's logo.

See the *Scoring* section for a breakdown of the marks awarded for different test cases.

Output

Your program must write several lines to standard output. The first line should contain a single integer S, the number of swaps in your solution.

Each of the S following lines will contain four integers: the coordinates (r_1, c_1) and (r_2, c_2) of two adjacent guides who will be asked to swap positions. The cell in the top-left corner has coordinates $(0, 0)$.

Sample Input

```
3 4
1 2 3 4
4 3 2 1
1 1 2 2
4 3 2 4
1 1 1 2
2 3 2 2
```

Sample Output

```
4
0 0 1 0
0 1 0 2
1 1 2 1
1 2 1 3
```

Performing the swaps in the sample output above gives the following grid:

```
4 3 2 4
1 1 1 2
1 3 2 2
```

Out of 12 cells, 11 match the sponsor's logo. This is an optimal solution.

Scoring

The score for each input scenario will be calculated based on the number of cells that match the desired logo, as follows:

- If the sequence of swaps is invalid (e.g. an invalid move is made or a guide is moved more than once) then the score for the test case will be 0%.
- If the number of cells that match is optimal (i.e., no other set of swaps gives a result with more matching cells), then the score for the test case will be 100%.
- If the number of cells that match is the same as, or less than, the original image, then the score for the test case will be 0%.
- Otherwise, let a be the number of matching cells in an optimal solution. Let b be the number of matching cells in the original input. Let x be the number of matching cells in your solution. The score for the test case (out of 100) will be computed by the formula:

$$100 \times \left(\frac{x-b}{a-b}\right)^6$$

For example, suppose that the optimal solution has $a = 100$ matching cells and the original input has $b = 50$ matching cells. Then scores would be assigned as follows:

Matching cells (x)	40	50	60	70	80	85	90	95	100
Score (%, rounded)	0	0	0	0	5	12	26	53	100

The judge's solution finds an optimal solution within the time and memory limits.

The following table lists the sizes of the inputs used in the test data.

R	C	Percentage of marks
1	100	10%
10	10	10%
20	20	10%
100	100	70%