Problem 1: Square Pasture

Farmer John has decided to update his farm to simplify its geometry. Previously, his cows grazed in two rectangular fenced-in pastures. Farmer John would like to replace these with a single square fenced-in pasture of minimum size that still covers all the regions of his farm that were previously enclosed by the former two fences.

Please help Farmer John figure out the minimum area he needs to make his new square pasture so that if he places it appropriately, it can still cover all the area formerly covered by the two older rectangular pastures.

INPUT FORMAT (file square.in):

The first line in the input file specifies one of the original rectangular pastures with four space-separated integers x_1, y_1, x_2, y_2, each in the range 0…10. The lower-left corner of the pasture is at the point (x_1,y_1), and the upper-right corner is at the point (x_2,y_2), where $x_2 > x_1$ and $y_2 > y_1$.

The second line of input has the same 4-integer format as the first line, and specifies the second original rectangular pasture. This pasture will not overlap or touch the first pasture.

OUTPUT FORMAT (file square.out):

The output should consist of one line containing the minimum area required of a square pasture that would cover all the regions originally enclosed by the two rectangular pastures.

SAMPLE INPUT:

6 6 8 8
1 8 4 9

SAMPLE OUTPUT:

49

In the example above, the first original rectangle has corners (6,6) and (8,8). The second has corners at (1,8) and (4,9). By drawing a square fence of side length 7 with corners (1,6) and (8,13), the original areas can still be enclosed; moreover, this is the best possible, since it is impossible to enclose the original areas with a square of side length only 6. Note that there are several different possible valid placements for the square of side length 7, as it could have been shifted vertically a bit.

Problem credits: Brian Dean

Problem 2: Block Game

Farmer John is trying to teach his cows to read by giving them a set of N spelling boards typically used with preschoolers ($1 \le N \le 100$). Each board has a word and an image on each side. For example, one side might have the word 'cat' along with a picture of a cat, and the other side might have the word 'dog' along with a picture of a dog. When the boards are lying on the ground, N words are therefore shown. By flipping over some of the boards, a different set of N words can be exposed.

To help the cows with their spelling, Farmer John wants to fashion a number of wooden blocks, each embossed with a single letter of the alphabet. He wants to make sufficiently many blocks of each letter so that no matter which set of N words is exposed on the upward-facing boards, the cows will be able to spell all of these words using the blocks. For example, if N=3 and the words 'box', 'cat', and 'car' were facing upward, the cows would need at least one 'b' block, one 'o' block, one 'x' block, two 'c' blocks, two 'a' blocks, one 't' block, and one 'r' block.

Please help the Farmer John determine the minimum number of blocks for each letter of the alphabet that he needs to provide, so that irrespective of which face of each board is showing, the cows can spell all N visible words.

INPUT FORMAT (file blocks.in):

Line 1 contains the integer N.

The next N lines each contain 2 words separated by a space, giving the two words on opposite sides of a board. Each word is a string of at most 10 lowercase letters.

OUTPUT FORMAT (file blocks.out):

Please output 26 lines. The first output line should contain a number specifying the number of copies of 'a' blocks needed. The next line should specify the number of 'b' blocks needed, and so on.

SAMPLE INPUT:

```
3
fox box
dog cat
car bus
```

SAMPLE OUTPUT:

```
2
2
2
1
0
1
1
0
0
0
0
0
0
0
2
```

0
0
1
1
1
1
0
0
1
0
0

In this example, there are N=3 boards, giving 2^3=8 possibilities for the set of upward-facing words:
fox dog car
fox dog bus
fox cat car
fox cat bus
box dog car
box dog bus
box cat car
box bat bus

We need enough blocks for each letter of the alphabet so that we can spell all three words, irrespective of which of these eight scenarios occurs.
Problem credits: Viktoriia Schwartz

Problem 3: The Cow-Signal

Bessie and her cow friends are playing as their favorite cow superheroes. Of course, everyone knows that any self-respecting superhero needs a signal to call them to action. Bessie has drawn a special signal on a sheet of M×N paper ($1 \le M \le 10, 1 \le N \le 10$), but this is too small, much too small! Bessie wants to amplify the signal so it is exactly K times bigger ($1 \le K \le 10$) in each direction. The signal will consist only of the '.' and 'X' characters.

INPUT FORMAT (file cowsignal.in):

The first line of input contains M, N, and K, separated by spaces.
The next M lines each contain a length-N string, collectively describing the picture of the signal.

OUTPUT FORMAT (file cowsignal.out):

You should output KM lines, each with KN characters, giving a picture of the enlarged signal.

SAMPLE INPUT:

```
5 4 2
XXX .
X. . X
XXX .
X. . X
XXX .
```

SAMPLE OUTPUT:

```
XXXXXX. .
XXXXXX. .
XX. . . . XX
XX. . . . XX
XXXXXX. .
XXXXXX. .
XX. . . . XX
XX. . . . XX
XXXXXX. .
XXXXXX. .
```

Problem credits: Nathan Pinsker