

## E. Garden

time limit per test: 2 seconds  
memory limit per test: 256 megabytes  
input: standard input  
output: standard output

Vasya has a very beautiful country garden that can be represented as an  $n \times m$  rectangular field divided into  $n \cdot m$  squares. One beautiful day Vasya remembered that he needs to pave roads between  $k$  important squares that contain buildings. To pave a road, he can cover some squares of his garden with concrete.

For each garden square we know number  $a_{ij}$  that represents the number of flowers that grow in the square with coordinates  $(i, j)$ . When a square is covered with concrete, all flowers that grow in the square die.

Vasya wants to cover some squares with concrete so that the following conditions were fulfilled:

- all  $k$  important squares should necessarily be covered with concrete
- from each important square there should be a way to any other important square. The way should go be paved with concrete-covered squares considering that neighboring squares are squares that have a common side
- the total number of dead plants should be minimum

As Vasya has a rather large garden, he asks you to help him.

### Input

The first input line contains three integers  $n$ ,  $m$  and  $k$  ( $1 \leq n, m \leq 100$ ,  $n \cdot m \leq 200$ ,  $1 \leq k \leq \min(n \cdot m, 7)$ ) — the garden's sizes and the number of the important squares. Each of the next  $n$  lines contains  $m$  numbers  $a_{ij}$  ( $1 \leq a_{ij} \leq 1000$ ) — the numbers of flowers in the squares. Next  $k$  lines contain coordinates of important squares written as " $x \ y$ " (without quotes) ( $1 \leq x \leq n$ ,  $1 \leq y \leq m$ ). The numbers written on one line are separated by spaces. It is guaranteed that all  $k$  important squares have different coordinates.

### Output

In the first line print the single integer — the minimum number of plants that die during the road construction. Then print  $n$  lines each containing  $m$  characters — the garden's plan. In this plan use character "X" (uppercase Latin letter X) to represent a concrete-covered square and use character "." (dot) for a square that isn't covered with concrete. If there are multiple solutions, print any of them.

### Examples

#### input

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

#### output

```
9
.X.
.X.
.XX
```

#### input

```
4 5 4
1 4 5 1 2
2 2 2 2 7
2 4 1 4 5
3 2 1 7 1
1 1
```

1 5  
4 1  
4 4

output

26  
X..XX  
XXXX.  
X.X..  
X.XX.