

L. Agricultural Archaeology

time limit per test: 2 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Recently Berland archaeologists found ancient text describing agricultural field of ancient people. It had a form of $n \times m$ rectangle divided into $n \cdot m$ unit cells. Each cell was sown with some kind of a food plant. There were 90 kinds of food plants popular in ancient Berland.

As written in the ancient text, region of each kind of food plant that was sown formed a single perfect square without any holes. Two square regions are adjacent if they share at least one unit cell border. The ancient text does not mention the sizes of the squares. But it provides something about the adjacent squares. For each square, it is known which kind of plants were sown to the each side of this square. Formally, for each food kind there are four lists: top neighbour kinds, right neighbour kinds, bottom neighbour kinds and left neighbour kinds. Food plants are written in arbitrary order in each list of neighbours.

Help archaeologists to reconstruct any possible agricultural field given the information from the ancient text.

Input

The first line contains three integer numbers n , m , k ($1 \leq n, m \leq 300, 1 \leq k \leq 90$) — the sizes of the field and the number of the sown kinds of food plants.

The sown food plants are encoded with characters which ASCII codes are from 33 ('!') to 122 ('z') inclusively.

Each of the following lines describes one sown food plant (square) and has the format: "char top right bottom left", where char is the character encoding the plant, and top, right, bottom, left are the strings of ASCII characters with codes from 33 to 122 — the lists of corresponding neighbours (all characters in each list are unique and written in arbitrary order). The empty list of neighbours is given by the character ' ' which ASCII code is 126. All characters which encode the sown food plants are different.

Output

Print the field in the form of $n \times m$ matrix of characters with ASCII codes from 33 to 122 — possible field corresponding to the given input. If there are many solutions, print any of them. It is guaranteed that at least one solution exists.

Examples

input
5 4 6 a {} b zc {} z ba {} {} dce e d z {} {} b {} {} z a c a z d {} d c z e {}
output
aabb aabb czzz dzzz ezzz