Puzzle

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	1024 megabytes

There is a grid with n rows and m columns. Each cell is either empty (denoted by `.`) or blocked (denoted by `#`).

You are also given k puzzle pieces, numbered $1, 2, \ldots, k$. Each piece is a 4-connected polyomino that cannot be rotated or flipped. Each piece is described by its minimum bounding rectangle, where `#` denotes a cell belonging to the piece and `.` denotes a cell not belonging to the piece.

You need to select some of the k pieces and place them on the grid such that:

- Each cell of each selected piece must cover exactly one empty cell of the grid.
- No two selected pieces cover the same cell.
- Every empty cell of the grid is covered by exactly one selected piece.

Count the number of valid arrangements. Two arrangements are considered different if and only if they use different sets of puzzle pieces, or there exists a piece whose covered cells differ between the two arrangements.

Input

There are multiple test cases. The first line of the input contains an integer T ($1 \le T \le 10^3$), indicating the number of test cases. For each test case:

The first line contains three integers n, m, and k ($1 \le n, m \le 8$, $1 \le k \le 8$), indicating the number of rows, columns, and puzzle pieces.

The next n lines each contain a string of length m, describing the grid. Here `.` denotes an empty cell and `#` denotes a blocked cell. It's guaranteed that the grid contains at least one empty cell.

Then k pieces are described. For each piece, the first line contains two integers r and c ($1 \le r, c \le 8$), indicating the number of rows and columns of its minimum bounding rectangle. The next r lines each contain a string of length c, describing the piece. It is guaranteed that each piece is a 4-connected polyomino, and that every row and every column of the bounding rectangle contains at least one `#`.

It is guaranteed that at most 10 test cases satisfy k greater than 5.

Output

For each test case, output one line containing an integer, indicating the number of valid arrangements.

Example

standard input	standard output
3 3 4 5 ..#. ..#. .... 3 2 .# .# ## 3 2 ## ## #. 3 2 .# .# ## 1 3 ### 2 1 # # 2 2 2 .. .. 1 1 # 1 2 ## 2 2 3 .. .. 1 1 # 1 1 # 1 2 ##	3 0 4

Note

The 3 arrangements of the first sample test case are described below, where the integers indicate the indices of the pieces.