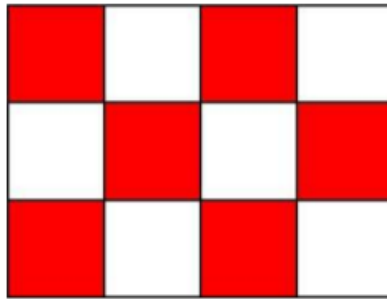


Checkerboard

Mirko has just installed a brand new drawing program. The program supports K different colours, denoted by integers from 1 to K . All drawing is performed on a canvas with dimensions $N \times N$. In the beginning, all cells are white (denoted by 1). The upper left cell of the canvas has coordinates $(0, 0)$. The first coordinate, x , increases iterating over rows, and the second, y , increases iterating over columns. Mirko's favourite pastime is drawing rectangular checkerboard patterns using the command `PAINT c x1 y1 x2 y2`, where c denotes the chosen colour, and $(x1, y1)$ and $(x2, y2)$ are coordinates of the upper left and lower right cells, respectively, of the rectangle being painted.

The upper left cell of the rectangle will be painted in the chosen colour, while the rest are determined by the checkerboard pattern. Cells that are not painted over by the chosen colour will retain their previous colour. For example, a white canvas painted over by a red checkerboard pattern will look like this:



Mirko has recently discovered two additional commands. He can save his painting at any time using the creatively named command `SAVE`, and load it again later using the command `LOAD x`, where x is a positive integer representing the ordinal number of the save.

Unfortunately, the program has crashed and Mirko's painting is lost forever. Luckily, Mirko has saved a log of all used commands. Can you help Mirko by reconstructing the lost painting?

Input

The first line of input contains three positive integers, N ($1 \leq N \leq 1000$), K ($2 \leq K \leq 100\,000$), and M ($1 \leq M \leq 100\,000$, M is the number of commands).

Each of the following M lines contains one of the three described commands. Input will not contain any illegal commands.

Output

Output must consist of N lines, each containing N integers representing the colours of cells in the corresponding row of the painting.

Sample input**Sample output**

4 3 2 PAINT 2 0 0 3 3 PAINT 3 0 3 3 3	2 1 2 3 1 2 1 2 2 1 2 3 1 2 1 2
3 3 4 PAINT 3 0 0 1 1 SAVE PAINT 2 1 1 2 2 LOAD 1	3 1 1 1 3 1 1 1 1
3 4 7 PAINT 2 0 0 1 1 SAVE PAINT 3 1 1 2 2 SAVE PAINT 4 0 2 0 2 LOAD 2 PAINT 4 2 0 2 0	2 1 1 1 3 1 4 1 3