Pizza

There is a small city of *n*m* blocks in the country named Hungrary. In some blocks there will be a pizzeria capable of providing pizza for the citizens living at a certain number of other blocks. A pizzeria can provide pizza for zero or more consecutive blocks directly to the north, zero or more consecutive blocks directly to the east and so on. Amount of pizza is precious, so there is just enough pizza to feed the citizens of the whole block. Therefore, it's crucial to open pizzerias so that every block is either a pizzeria or fed by exactly one pizzeria. Here is an example. Pizzerias are represented in the form **id(PizzaCapacity)**. For example, the 4-th pizzeria is capable of providing pizza for 5 other blocks.

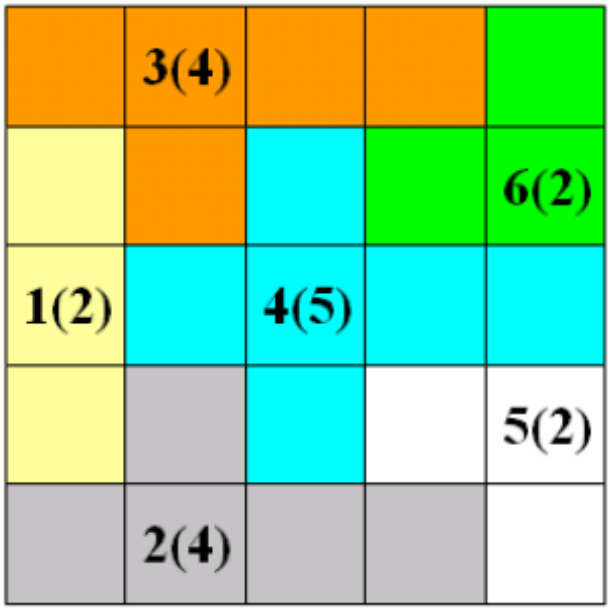

Fig 1. A pizzeria system

Input

The input consists of at most 50 test cases. Each case begins with a line containing three non-negative integers *n*, *m* and *k* (1 <= *n*,*m* <= 30, 1 <= *k* <= 200), where *k* is the number of pizzerias. In the following *k* lines pizzerias are described, by three non-negative integers *x*, *y*, *c*(1 <= *x* <= *n*, 1 <= *y* <= *m*), the coordinate (west-south corner is (1,1)) and pizza capacity, one line for each pizzeria. It is guaranteed that *nm* - *k* = *c*1 + *c*2 + ... + *ck* (just enough pizza). The last case is followed by a single zero, which should not be processed.

Output

For each test case, print the case number in the first line, then *k* lines followed, one for each pizzeria. Each pizzeria is described by four non-negative integers *n*, *e*, *s*, *w*, the amount of blocks fed in each direction. It is guaranteed that at least one solution exists. Print a blank line after each test case.

Sample Input

2 2 2
1 1 1

2 2 1
5 5 6
1 3 2
2 1 4
2 5 4
3 3 5
5 2 2
5 4 2
0

Output for the Sample Input

Case 1:
1 0 0 0
0 0 1 0

Case 2:
1 0 1 0
1 2 0 1
0 2 1 1
1 2 1 1
0 0 1 1
1 0 0 1