PROBLEM DESCRIPTION:

A car dealer has a set of $\mathcal{C}$ cars to be placed in $\mathcal{P}$ consecutive parking slots (with $\mathcal{C} < \mathcal{P}$). According to local regulations, all cars are ranked by a third-party company in $\mathcal{A}$ different attributes so that the customers have an unbiased evaluation of the car characteristics. The goal of the car dealer is to place each car in a slot, but he has some restrictions to consider.

First of all, due to their size, some cars cannot be parked in some concrete slots. Moreover, due to selling strategies the car dealer has some constraints of his own:

- At most two consecutive cars of the same brand are allowed. This is because otherwise the customer could easily compare those cars and determine which one is the best choice. Hence, some cars would never be sold.
- Two cars cannot be next to each other if the best attribute of one of the cars happens to be the worst attribute of the other one. In this case the car with that attribute being its worst one would create a poor impression to the customer.

The goal of this exercise is, using SAT technology, to find an appropriate distribution of the cars so that all constraints are satisfied. In order to encode the problem, it might be helpful to consider that $\mathcal{A} << \mathcal{C}$ and, if $\mathcal{B}$ is the number of different brands, also $\mathcal{B} << \mathcal{C}$.

INPUT FORMAT:

You will be given a C++ program `generator.cpp` that, given 4 integers $(\mathcal{C}, \mathcal{P}, \mathcal{B}, \mathcal{A})$, outputs a problem in the following format:

- First line is

  $\mathcal{C}$ $\mathcal{P}$ $\mathcal{B}$ $\mathcal{A}$ 0

- Then for each car we have a line of the form:

  `brand best_attribute worst_attribute [list of forbidden slots]` 0

  where `[list of forbidden slots]` can be empty.

Example:

```
6 10 2 3 0
1 3 1 1 2 5 0
2 2 3 0
1 3 1 1 3 0
1 1 3 2 8 0
1 1 2 1 10 0
2 2 1 4 10 0
```

Represents an instance with 6 cars, 10 parking slots, 2 brands and 3 attributes.

Car c_1 is of brand a_1, has a_3 as its best attribute, a_1 as its worst one and cannot be parked in slots s_1, s_2 and s_5.

Car c_2 is of brand a_2, has a_2 as its best attribute, a_3 as its worst one and can be parked anywhere.

OUTPUT FORMAT:

The distribution will be printed in a single line ending with zero:

```
4 2 x 5 1 9 10 3 6 7 8 0
```

where in this case car c_4 is in the first slot and, for example, no car is parked in the third slot.