

## Problem Description

A set of  $n^2$  computer scientists went to the movies. Fortunately, the theater they chose has a square layout:  $n$  rows, each one with  $n$  seats. However, these scientists are not all from the same research area and they want to seat together. Indeed, there are  $K$  independent research groups of scientists among them (no scientist belongs to two of them) with a distinguished leader for each group. Then the leader bought the tickets for his whole group, and he did it in such a way that all his group could seat occupying a rectangular set of seats (and everyone in this set of seats belongs to the same group). Every group was placed satisfying this bizarre condition, although the scientists did not care where the actual assigned areas were. The usher was informed of the situation and he decided to annotate in a theater map a satisfactory seats deploying. He thought that if he wrote the position of each group's leader in the map indicating besides the corresponding group size, he could tell where to accommodate every scientist. But he discovered that it is not so easy! The usher asks for your help. You must tell him a way to place the  $K$  rectangular areas with the given sizes, and with the corresponding leader for each group seated where it was originally assigned.

### Input

Input consists of several test cases, each one defined by a set of lines:

- the first line in the case contains two numbers  $n$  and  $K$  separated by blanks, with  $n$  representing the size of the theater ( $0 < n < 20$ ) and  $K$  the number of groups ( $K \leq 26$ );
- the next  $n$  lines describe the usher's map. A one-digit decimal number in the map indicates the seat of a leader and the size of his group. A point indicates that no leader will sit there.

The end of the input is indicated by the line

0 0

### Output

For each test case, display an answer consisting in  $n$  lines each one of them with  $n$  characters representing a seat occupation for the theater. Each group is assigned to an uppercase letter and all of its members are identified with that letter. No two groups are assigned to the same letter.

### Sample Input

```
3 3
3.4
...
.2.
7 18
...4.2.
...45..
222..3.
```

...2..3  
.24...2  
...2.3.  
22..3..  
0 0

## Sample Output

ABB  
ABB  
ACC  
AAAABCC  
DDDDBEF  
GHIIBEF  
GHJKB EF  
LLJKBMM  
NOJPQQQ  
NOJPRRR