UVa 10004 Bicoloring

In 1976 the ``Four Color Map Theorem" was proven with the assistance of a computer. This theorem states that every map can be colored using only four colors, in such a way that no region is colored using the same color as a neighbor region.

Here you are asked to solve a simpler similar problem. You have to decide whether a given arbitrary connected graph can be bicolored. That is, if one can assign colors (from a palette of two) to the nodes in such a way that no two adjacent nodes have the same color. To simplify the problem you can assume:

- no node will have an edge to itself.
- the graph is nondirected. That is, if a node a is said to be connected to a node b, then you must assume that b is connected to a.
- the graph will be strongly connected. That is, there will be at least one path from any node to any other node.

Input

The input consists of several test cases. Each test case starts with a line containing the number n ($1 < n < 200$) of different nodes. The second line contains the number of edges l. After this, l lines will follow, each containing two numbers that specify an edge between the two nodes that they represent. A node in the graph will be labeled using a number a ($0 \leq a < n$).

An input with $n = 0$ will mark the end of the input and is not to be processed.

Output

You have to decide whether the input graph can be bicolored or not, and print it as shown below.

Sample Input

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

Sample Output

```
NOT BICOLORABLE.
BICOLORABLE.
```

Uva 532 Dungeon Master

You are trapped in a 3D dungeon and need to find the quickest way out! The dungeon is composed of unit cubes which may or may not be filled with rock. It takes one minute to move one unit north, south, east, west, up or down. You cannot move diagonally and the maze is surrounded by solid rock on all sides.

Is an escape possible? If yes, how long will it take?

Input Specification

The input file consists of a number of dungeons. Each dungeon description starts with a line containing three integers *L*, *R* and *C* (all limited to 30 in size).

L is the number of levels making up the dungeon.

R and *C* are the number of rows and columns making up the plan of each level.

Then there will follow *L* blocks of *R* lines each containing *C* characters. Each character describes one cell of the dungeon. A cell full of rock is indicated by a `#' and empty cells are represented by a `.'. Your starting position is indicated by `S' and the exit by the letter 'E'. There's a single blank line after each level. Input is terminated by three zeroes for *L*, *R* and *C*.

Output Specification

Each maze generates one line of output. If it is possible to reach the exit, print a line of the form

```
Escaped in x minute(s).
```

where *x* is replaced by the shortest time it takes to escape.

If it is not possible to escape, print the line

```
Trapped!
```

Sample Input

```
3 4 5
S....
.###.
.##..
###.#

#####
#####
##.##
##...

#####
#####
#.###
####E

1 3 3
S##
#E#
###

0 0 0
```

Sample Output

```
Escaped in 11 minute(s).
Trapped!
```

POJ 3617 Best Cow Line

Time Limit:1000MS Memory Limit:65536K

Description

FJ is about to take his N ($1 \leq N \leq 2{,}000$) cows to the annual"Farmer of the Year" competition. In this contest every farmer arranges his cows in a line and herds them past the judges.

The contest organizers adopted a new registration scheme this year: simply register the initial letter of every cow in the order they will appear (i.e., If FJ takes Bessie, Sylvia, and Dora in that order he just registers BSD). After the registration phase ends, every group is judged in increasing lexicographic order according to the string of the initials of the cows' names.

FJ is very busy this year and has to hurry back to his farm, so he wants to be judged as early as possible. He decides to rearrange his cows, who have already lined up, before registering them.

FJ marks a location for a new line of the competing cows. He then proceeds to marshal the cows from the old line to the new one by repeatedly sending either the first or last cow in the (remainder of the) original line to the end of the new line. When he's finished, FJ takes his cows for registration in this new order.

Given the initial order of his cows, determine the least lexicographic string of initials he can make this way.

Input

* Line 1: A single integer: N
* Lines 2..N+1: Line i+1 contains a single initial ('A'..'Z') of the cow in the ith position in the original line

Output

The least lexicographic string he can make. Every line (except perhaps the last one) contains the initials of 80 cows ('A'..'Z') in the new line.

Sample Input

```
6
A
C
D
B
C
B
```

Sample Output

```
ABCBCD
```