# 3 Flood-It!

Crazyman loves the Google+ game Flood-It. However, his alter ego Lazyman does not want to spend long hours thinking about the best solution for the game. Hence he has decided to use the students at IOITC as slave labour to write the best programs which will make him a Super-duper-expert-pixie in the game.
The game consists of an $N \times N$ grid with each cell initially coloured with one of six colours (red, blue, green, yellow, pink, and violet - denoted by A,B,C,D,E,F in this problem). In each step, you can choose one of the six colours to colour the top left cell. This colour spreads to all connected cells of the same colour as the original colour of the top-left cell - similar to the floodfill tool in paint software. Formally, the process can be described as such : Two cells in the board are said to be adjacent if they share an edge and have the same colour. A set of cells is called a connected component if it is possible to move between any pair of cells in the set via a path of adjacent cells alone, and no further cell can be added to the set satisfying this property. When the top left corner is about to be coloured by some new colour, the entire connected component containing the top left cell changes to the new colour. The game is complete when all cells have the same colour.
A video showing the original game is available, with filename `demo.ogg`. Ignore the limit of 25 moves shown in the video, as it is irrelevant for your task.
You will be given 10 inputs, each corresponding to a Flood-It board. For each input, you have to produce an output, consisting of a sequence of moves which completes the game. The fewer moves you use, the better your solution (and the better it will be for you as Crazyman/Lazyman plans to give you a mild electric shock for each move you make.)
*This is an output-only task. You do not need to submit any program.*

## Input format

- Line 1 : A single integer $N$, the number of rows and columns in the grid.
- Lines 2 to $N+1$ : Each line has $N$ characters, describing a row of the grid, from left to right.

(The top-left cell corresponds to the first position in line 2.)

## Output format

The output should consist of a single line with a sequence of *at most 2500* characters, the sequence of colours chosen for the top-left cell.

## Sample Input

```
4
ABAF
CCCA
BBAB
DCDE
```

## Sample Output

```
BAFACBADCE
```

## Scoring

Every test case is worth 10 marks. Each test case is scored independently of the others. For each test case, your score is computed as follows : If the number of moves is more than 2500, or if the sequence of moves does not correctly result in all cells having the same colour, the score

is 0. Otherwise, your score for the test case is at least 1. The score will be computed as some monotonically decreasing function of the number of moves used, based on comparing all solutions (the participants' solutions as well as the judges' solutions). On each test case, the best solution amongst all the participants' and judges' solutions will get 10 marks.
Your final score for the task will be the sum of the scores for the 10 test cases.

## Playing the game

You are provided with an executable `play` which you can use to play the game. With `play` in the current directory, run `play name_of_input_file` to play the game on a particular input.

## Submitting your solution

Your output files should have the same names as the input files, with `.in` replaced by `.out`. To submit your solutions, put all the output files you wish to submit in a particular directory, and in that directory, run

```
zip floodit.zip *.out
```

to create a file `floodit.zip` with your output files. Submit this file to the grader. The solutions will be evaluated after the test. Ignore the tokens for this problem.