

CS/240/Project/?

DUE: Some point, I don't know, we'll let you know, just work on it

Copy the initial files, from

```
/homes/cs240/somelab/lab.tar.gz
```

you know the deal.

1 Introduction

The goal of this lab is to work on problem solving skills, as well as translating the algorithm into code. The idea is that by understanding and analyzing the game and its rules, you will be able to implement a solver for the game 0hh1 which can:

- Take in a board of a given size,
- Determine the unique solution of the board
- Print the completed board

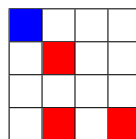
2 Game Rules

0hh1 is a game played in a square board with a even dimension(usually from 4 and 10, though larger boards are possible) with red and blue tiles in which, similar to sudoku, the player must fill in the board by following a few simple rules:

1. There cannot be three of the same color next to each other in a row
2. There cannot be three of the same color next to each other in a column
3. There must be an equal number of each color in every row
4. There must be an equal number of each color in every column
5. No two rows can be the same
6. No two columns can be the same

3 Game example

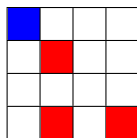
Let's look at a simple 4x4 board to get a feel for how the game works:



3.1 Three in a row/column

The first, arbitrarily chosen, rule we want to check is that there are no places where three of the same color are together.

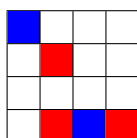
We have this board:



Let's look at the last row:



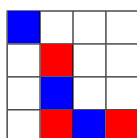
Notice there are two of the same color separated by a blank square. Therefore, the square in between them must be blue. Now we have:



But if we look at the second column:

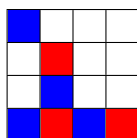


We see that there is another place which there is only one possibility. Thus:

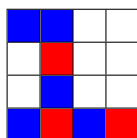


3.2 Equal Numbers

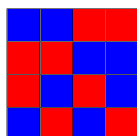
Next, we will make sure that the rows and columns have an equal number of each color. We have a few spaces we can fill because there are already enough of one color. First, we fill in the last row:



Next, the second column:

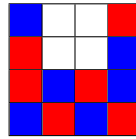


This opens up many spaces we can now fill the same way to finish this board:

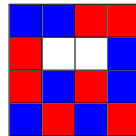


3.3 No two rows or columns are the same

The 4x4 board used in the previous example was not complicated enough to need to check for this rule, but this is an example of when it may be used:



The first and last rows have the same ends, but the last row is filled in. By this rule, the first row must be:



Then the other two spaces can be filled in by rule 2.

NOTE: Not every board can be solved by going through each function only once, especially as the boards grow larger.

Your program should be called `0hh1` and should have the following call signature:

`0hh1`

4 Format

You may solve the problem however you wish, but, for the sake of testing, the format of the input and output should follow the standards as follows:

4.1 Input

The input given to your program will come from stdin and will be of the form:

B
R

R R

Containing B for Blue, R for Red, and a space for empty.

4.2 Output

Your output should include, but is not necessarily limited to, the final board of the form:

```
+++-----+
|B|B|R|R|
+++-----+
|R|R|B|B|
+++-----+
|R|B|R|B|
+++-----+
|B|R|B|R|
+++-----+
```

You may have more in your output if you wish, but your output **MUST**, at least, contain the board in the correct format.

NOTE: Any additional output is up to you, and should be useful to the user, so you should still remove print statements used for debugging,

Turning in

Please submit your project directory in the usual fashion:

```
turnin -c cs240 -p ...
```