UVa Online Judge

13144 Go Boards

Go is an ancient Chinese game which is played in a board with a grid of 19×19 lines. Two players can place white or black "stones" in the intersections of the lines, and there are some simple rules to capture the opponent's pieces by surrounding them. Only recently, in 2016, a computer program has been able to beat a leading professional go player. Also in 2016, the number of legal positions in a 19×19 go board has been calculated and it is a bit larger than 2.08×10^{170}.

That number represents less than 1.2% of the ways that you can setup a go board if you are allowed to place as many black or white stones in as many (or as few) intersections of the board, regardless of the legality of the resulting position according to the actual rules of go. I have decided to count that number exactly, and for doing that I have a go board and a lot of stones of each color. My plan is to produce each of the possible positions following a systematic order to avoid repeating any.

A sample go goard of 19×19 lines. (Extracted from Wikimedia Commons)

Five months ago, I started with an empty board. I took a photo of it, and called it "Position number 1". Then, I placed a white stone in the top-left intersection, took another photo, and called it "Position number 2". Then, I removed the white stone and placed a black one in its place, and that was "Position number 3". Then, I removed the black stone and put a white stone again, but now in the position just to the right of the top-left, and I named the corresponding photo as "Position number 4". Positions 5 and 6 were like position 4 but with an additional white or black stone in the top-left position, respectively. At this point I noticed that taking photos was not really necessary and that skipping that step would make the process faster, so I stopped. For position 7, I removed the two stones and placed a single black one in the intersection just to the right of the top-left. Position 8 was like position 7 but with an additional white stone in the top-left position, and position 9 had two black stones, one in the top-left intersection and another in the one to its right. Positions 10 to 18 were like positions from 1 to 9 but with an additional white stone in the second position to the right of the top-left, and positions 19 to

27 were also like position 1 to 9 but now with a black stone in the second position to the right of the top-left. As you may guess by now, positions 28 to 54 were like positions 1 to 27 but with an additional white stone in the intersection three places to the right of the top-left, and positions 55 to 81 were like 1 to 27 with a black stone in the same place. If I follow with this method filling the intersections from left to right and from top to bottom I will be able to produce all positions without any repetition. I will finish when all intersections are filled with black stones.

My plan was proceeding well for some time. I was working 8 hours per day and 6 days each week and after some practice I was able to produce a position every 3 seconds on average. Two days ago, I had arrived to position 834142 when I accidentally dropped the board to the floor. Now I cannot continue with position 834143 because I don't remember the exact positions of the stones in 834142. I don't want to start again from the first position. Also, even if I start from the beginning, something similar may happen again in the future even when I would be much further along the project. I need a program to help me to restart my count at any given position by drawing the corresponding board.

Input

The input format is as follows:

An integer in a single line which says the number of problems to solve. Then, for each problem, a line containing just the number of the board that should be drawn.

Output

For each problem, the corresponding go board should be drawn. There must be 19 lines with 19 characters each, representing each intersection of the board. Each character can be either a '`.`' (period) for an empty intersection, a '`W`' for a white stone, or a '`B`' for a black stone. An empty line should be printed after each board (also after the last one).

Sample Input

```
3
5
834142
987566
```

Sample Output

```
WW.................
...................
...................
...................
...................
...................
...................
...................
...................
...................
...................
...................
...................
...................
...................
...................
...................
...................
...................
```

```
..................
..................
..................

.W..B.W.W.BWW......
..................
..................
..................
..................
..................
..................
..................
..................
..................
..................
..................
..................
..................
..................
..................

WWW..BWW.BWBW......
..................
..................
..................
..................
..................
..................
..................
..................
..................
..................
..................
..................
..................
..................
..................
..................
..................
..................
```