6. TIC-TAC-LOGIC

PROBLEM: The Tic-Tac-Logic game, also known as Binairo, is played on an *N* by *N* grid that starts out by being partially filled with Xs and Os. The goal is to place Xs and Os in all of the empty cells of the grid using the following rules:

1. Each row and each column must contain an equal number of Xs and Os.
2. There are no more than two consecutive Xs or Os in any row or column.
3. Each row is unique among rows. Each column is unique among columns.

EXAMPLE: The diagram at the left is an initial grid; the diagram at the right is the completed grid. These diagrams correspond to the first line of Sample Input below:

		o	x		o
			x		
o	x		x		
		o			
					o

x	x	o	x	o	o
o	o	x	o	x	x
x	o	x	x	o	o
o	x	o	x	o	x
o	x	o	o	x	x
x	o	x	o	x	o

INPUT: You will be given 10 games to play. Each game is an integer *N* representing the size of the grid ($N <= 20$), followed by a string describing the initial contents of the grid. The string will represent the grid, row by row, from top to bottom, with empty cells run-length encoded. The numbers and the strings in the input are separated by white space (spaces, tabs, and/or newline characters).

Here is the example board above (a dash represents an empty cell) before the run-length encoding:

```
--OX-O---------X--OX-X----O--------O
```

And here is the run-length encoding of the empty cells: **`2OX1O9X2OX1X4O8O`**

OUTPUT: For each input game, print the hex encoding of the major diagonal (top left to bottom right). This is done in three steps. First, create a string of X's and O's from the major diagonal. Second, replace each X by the number 1, each O by the number 0, and consider the string as a binary number. Third, convert this binary number to hexadecimal. Use capital letters A, B, C, ...F. Remove all leading zeros in the hex encoding.

In the example above, the major diagonal is **`XOXXXO`**. This converts to 101110, which is 2E in hex.

6. TIC-TAC-LOGIC

SAMPLE INPUT: (*We are showing data for 3 games; the Test Data will have 10 games.*):

```
6 2OX1O9X2OX1X4O8O
8 2O3O2X9O2XXX3X5O3O5XX1X1X9X1X2
10 X3O2O4OO9O2O4X3X12X2O5X5X1OO3O9O3XX1O1O2XX2O3
```

SAMPLE OUTPUT: (*Output must match exactly. Same capitalization.):*

1. 2E
2. E6
3. 3F2

6. TIC-TAC-LOGIC

TEST DATA

TEST INPUT:

1. 6 11XO1O6X1OX5X5
2. 8 1X4X5X1X3O8O2O6XX1X3O5O2X3XO2O2
3. 8 1X12X6O4O7OO1X3X11O1O3O2X1
4. 10 5O2O1O2X8X3O7XX2X1OX8O3OO2O3OX7X5O2X4X5X2O4X
5. 10 X2X1O1XX6O3O2X14X5XX2XO2X2O5X4X4X4X7X4O2O8
6. 10 3X1X3X2X5X12XX5X1O3O3XO3X8X3O2O3X2O2O2O11O3XO
7. 14
   3O1X3XX6X10O6OO1OO2O10O5O3O5X1XX6O20OO5XX1O5X1O1OO2O4O10O3O9O2O
   3XX3X3X2XX2X6X4X6
8. 14
   OX1X4O2XO15O1X2O2OO2O6O3OO2X7X3X18XX5XX4XX2X3X5X1X2OO4O10O8XO9O
   X5O2O6X1XX1O10X7
9. 12 2O5O4X1X7X1X3O6O2X1O15O8X1X1X8O2X30O3O5X4O1X3O3X1O1X1
10. 12 XX5O2X2X4O3OO15X5X4OX1O2XX7O4O3X11XX1O2X17X1OX18X7X3

TEST OUTPUT:

1. 7
2. 1
3. BF
4. 3A1
5. 3A1
6. 4
7. E50
8. DA2
9. F3A
10. F0F