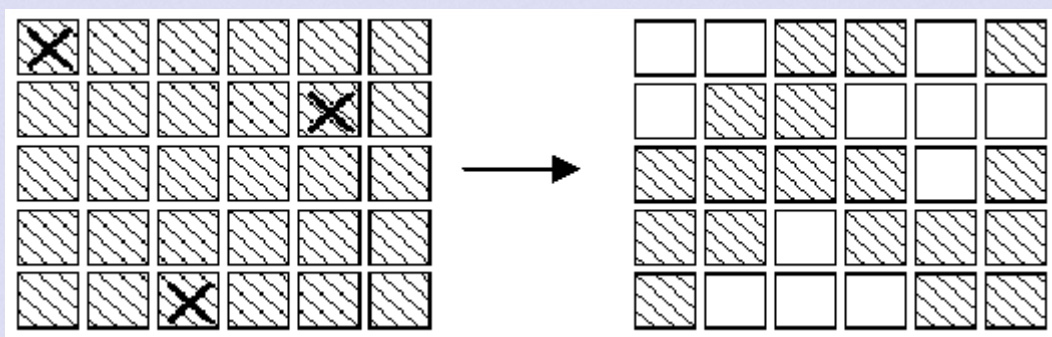


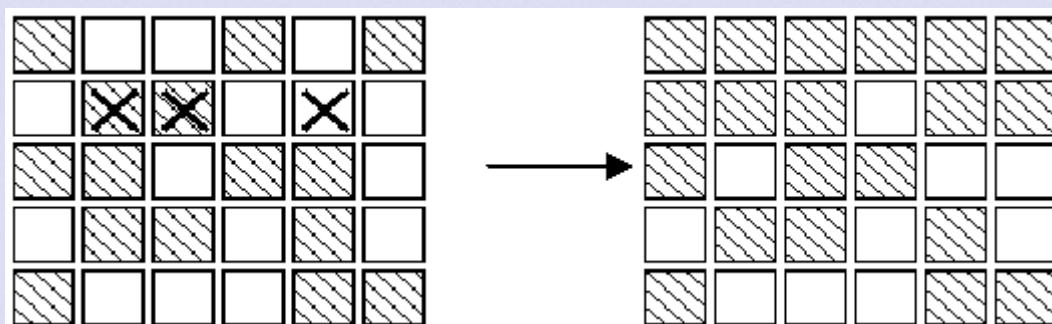
熄灯问题

(lamp.cpp/.c)

有一个由按钮组成的矩阵，其中每行有6个按钮，共5行。每个按钮的位置上有一盏灯。当按下一个按钮后，该按钮以及周围位置(上边、下边、左边、右边)的灯都会改变一次。即，如果灯原来是点亮的，就会被熄灭；如果灯原来是熄灭的，则会被点亮。在矩阵角上的按钮改变3盏灯的状态；在矩阵边上的按钮改变4盏灯的状态；其他的按钮改变5盏灯的状态。



在上图中，左边矩阵中用X标记的按钮表示被按下，右边的矩阵表示灯状态的改变。对矩阵中的每盏灯设置一个初始状态。请你按按钮，直至每一盏等都熄灭。与一盏灯毗邻的多个按钮被按下时，一个操作会抵消另一次操作的结果。在下图中，第2行第3、5列的按钮都被按下，因此第2行、第4列的灯的状态就不改变。



请你写一个程序，确定需要按下哪些按钮，恰好使得所有的灯都熄灭。

根据上面的规则，我们知道1) 第2次按下同一个按钮时，将抵消第1次按下时所产生的结果。因此，每个按钮最多只需要按下一次；2) 各个按钮被按下的顺序对最终的结果没有影响；3) 对第1行中每盏点亮的灯，按下第2行对应的按钮，就可以熄灭第1行的全部灯。如此重复下去，可以熄灭第1、2、3、4行的全部灯。同样，按下第1、2、3、4、5列的按钮，可以熄灭前5列的灯。

Input (lamp.in)

第一行是一个正整数 N，表示需要解决的案例数。每个案例由5行组成，每一行包括6个数字（0或1）。相邻两个数字之间用单个空格隔开。0表示灯的初始状态是熄灭的，1表示灯的初始状态是点亮的。

Output (lamp.out)

对每个案例，首先输出一行，输出字符串“PUZZLE #m”，其中m 是该案例的序号。接着按照该案例的输入格式输出5行，每一行包括6个数字（0或1）。相邻两个数字之间用单个空格隔开。其中的1表示需要把对应的按钮按下，0则表示不需要按对应的按钮。

Sample

输入样例	输出样例
2	PUZZLE #1
0 1 1 0 1 0	1 0 1 0 0 1
1 0 0 1 1 1	1 1 0 1 0 1
0 0 1 0 0 1	0 0 1 0 1 1
1 0 0 1 0 1	1 0 0 1 0 0
0 1 1 1 0 0	0 1 0 0 0 0
0 0 1 0 1 0	PUZZLE #2
1 0 1 0 1 1	1 0 0 1 1 1
0 0 1 0 1 1	1 1 0 0 0 0
1 0 1 1 0 0	0 0 0 1 0 0
0 1 0 1 0 0	1 1 0 1 0 1
	1 0 1 1 0 1