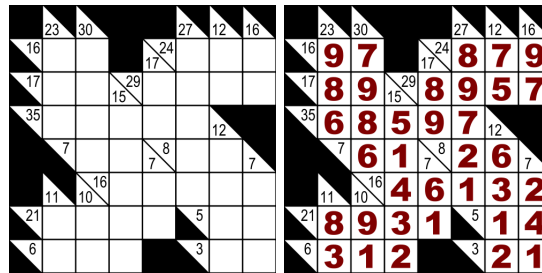


首先介绍一下 Kakuro(カックロ) 这个游戏。

游戏规则为：

- 方形空格中填入 1 ~ 9 的整数。
- 被斜线分开的方格中，右上角的数字等于其右侧邻接之连续方格中数字之和，左下角的数字等于其下方邻接之连续方格中数字之和。
- 无论是横向还是纵向，连续方格中的数字不能重复。



左边为一个 Kakuro 游戏，右边为这个游戏的唯一解。

我们称一开始给出的数字为**线索**，称需要填入数字的地方为**空格**。如果一个格子包含线索那么就不需要填入数字。我们约定所有的谜题都非空，即至少有一个空格需要被填入。

注意：在以下题目中的游戏规则可能会有所不同，请认真阅读在每个题目下的规则。

注意事项

- 比赛与评测均在 Windows 下。评测的时候使用 lemon 评测，请大家在选手目录下直接放三个文件，不需要子文件夹。
- 在原则上，编译会开 O2,C++11 和放大栈空间限制。具体会在考试开始后通知大家。

Problem B. Cross sum

输入文件: cross.in
输出文件: cross.out
时间限制: 2 second
空间限制: 512 megabytes

游戏规则：

- 空格中填入**正整数**。
- 被斜线分开的方格中，右上角的数字等于其右侧邻接之连续方格中数字之**异或和**，左下角的数字等于其下方邻接之连续方格中数字之**异或和**。
- **所有空格中填入的整数都不能重复**。

Apia 给了 Rimbaud 一个 Kakuro 谜题。心不灵手不巧的 Rimbaud 根本不会做 Kakuro，所以她请求你帮她解决。由于 Rimbaud 很年幼，只会对不超过 $2^{60} - 1$ 的数字进行运算。**所以希望谜题的解中每个数字都不超过 $2^{60} - 1$ 。**

Input

每组数据包含多组测试数据。第一行包含一个整数 T 表示测试数据组数。

对于每组测试数据，第一行，两个正整数表示 n, m 表示这个游戏的行和列。

接下来 n 行，每行包含 m 个 0 到 4 的数字，第 i 行第 j 列表示第 i 行第 j 列格子的种类。

- 0 表示这个格子既不是空格也不是线索。
- 1 表示这个格子左下角包含线索，右上角没有线索。
- 2 表示这个格子右上角包含线索，左下角没有线索。
- 3 表示这个格子左下角右上角都包含线索。
- 4 表示这个格子为空格。

输入保证这个从格式上来说一定是个合法的 Kakuro 谜题，即每一段连续的空格的左边或者上面的格子包含线索。

接下来 n 行，每行包含若干个**非负整数**，按从左往右的顺序给出谜题中的每个线索。特别地如果这个格子的种类为 3，那么先给出左下角的线索，再给出右上角的线索。

Output

对于每组测试数据，如果有解，那么输出 n 行，每行按从左往右的顺序输出往空格中填入的数，要求空格内填入的数字在 1 到 $2^{60} - 1$ 之间。否则输出一个 -1 。

Example

cross.in	cross.out
3	
3 3	1 3
0 1 1	2 4
2 4 4	-1
2 4 4	-1
3 7	
2	
6	
3 3	
0 1 1	
2 4 4	
2 4 4	
1 1	
1	
1	
2 2	
0 1	
2 4	
0	
0	

Constraints

对于 10% 的数据, 保证 $n, m \leq 3$ 。

对于 30% 的数据, 保证 $n, m \leq 15$ 。

对于 50% 的数据, 保证 $n, m \leq 40$ 。

对于另外 20% 的数据, 保证只有第一行第一列包含线索, 剩下的地方全都是空格。

对于 100% 的数据, 保证 $3 \leq n, m \leq 200, T \leq 5$, 保证初始局面中的每个数字不超过 $2^{60} - 1$ 。