

## Problem A. Determinant

输入文件: `stdin`  
输出文件: `stdout`  
时间限制: 5 second  
空间限制: 1024 megabytes

你有一个  $n \times n$  的矩阵，一开始里面都是 0。你有  $n$  个操作，对于每个操作会选择一个左下角为  $(x_1, y_1)$  右上角为  $(x_2, y_2)$  的子矩形，在这个矩形中独立随机等概率地选择一个位置，将这个位置中的元素加上 1。

最后这个矩阵的行列式的期望值，答案对  $10^9 + 7$  取模。

### Input

第一行一个整数  $T$ ，表示有  $T$  组数据。

每组数据，第一行一个整数  $n$ ，接下来  $n$  行每行四个整数  $x_1, x_2, y_1, y_2$ 。

### Output

每组数据一个整数表示答案，答案对  $10^9 + 7$  取模。

### Example

stdin	stdout
2	1000000006
2	277777778
2 2 1 1	
1 1 2 2	
3	
1 1 1 1	
1 2 1 2	
1 3 1 3	

### Constraints

- 对于 10% 的数据,  $n \leq 3$ 。
- 对于另外 10% 的数据, 每个矩形满足  $x_1 = x_2, y_1 = y_2$ 。
- 对于另外 10% 的数据, 每个矩形满足  $x_1 = x_2, n \leq 100$ 。
- 对于另外 20% 的数据, 每个矩形满足  $x_1 = x_2$ 。
- 对于另外 20% 的数据,  $n \leq 100$
- 对于 100% 的数据,  $n \leq 10^5$ , 满足  $1 \leq x_1 \leq x_2 \leq n, 1 \leq y_1 \leq y_2 \leq n, T \leq 10$ 。

## Problem B. String

输入文件: `stdin`  
输出文件: `stdout`  
时间限制: 2 second  
空间限制: 1024 megabytes

大家都很喜欢字符串和 Thue-Morse 序列。Thue-Morse 序列  $\{T_i\}$  可以这样定义，如果  $i$  的二进制表示中 1 的个数为偶数，那么  $T_i = 0$ ，否则  $T_i = 1$ 。

定义  $k$  阶 Thue-Morse 序列  $M_k$  为取整个序列中前  $2^k$  项得到的字符串，现在将进行后缀排序之后，问第  $x$  小的后缀是哪一个？

比如  $M_3 = 01101001$ ，那么后缀从小到大分别是 5, 6, 3, 0, 7, 4, 2, 1。

### Input

第一行两个正整数  $k, q$ ，其中  $q$  表示询问个数。

接下来一行  $q$  个整数，表示  $q$  个询问，询问的大小不超过  $2^k$ 。

### Output

输出一行， $q$  个整数，表示答案。

### Example

stdin	stdout
3 8 1 2 3 4 5 6 7 8	5 6 3 0 7 4 2 1

### Constraints

对于 30% 的数据，保证  $k \leq 18$ 。

对于 50% 的数据，保证  $k \leq 23$ 。

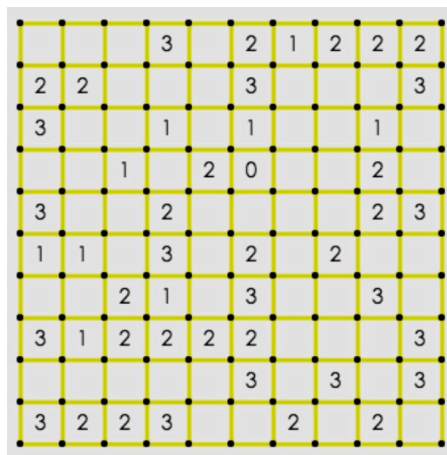
对于 100% 的数据，保证  $1 \leq k \leq 60, q \leq 10^5$ 。

## Problem C. loopy

输入文件: loopy\*.in

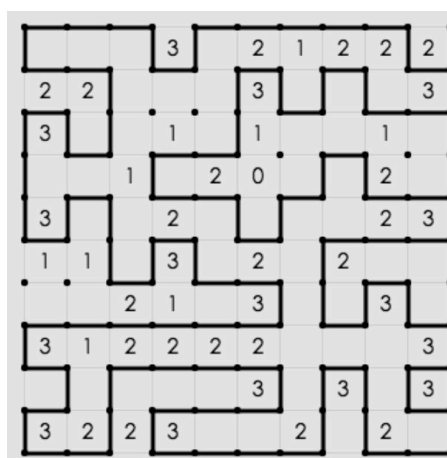
输出文件: loopy\*.out

这是一道提交答案题。



你要在从网格图的边上选出一些边使得这些边形成一个简单环。这个网格图的有些格子中的数字，数字表示这个格子周围有几条边被选中了。

比如说上面的游戏完成后张这样：



数据保证解是唯一的。

### Input

第一行一个正整数  $n$ ，表示网格图的边长。

接下来  $n$  行，每行  $n$  个整数，如果是  $-1$  表示这个格子没有限制，否则表示这个格子周围被选的边数。

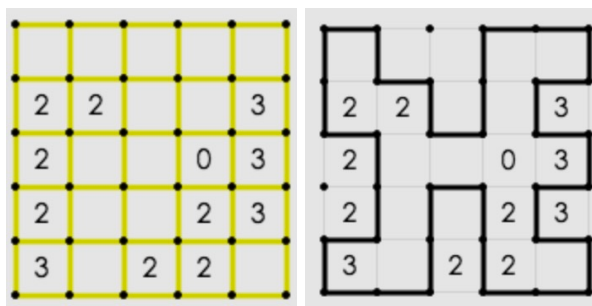
### Output

第一行一个数  $m$ ，表示环长，接下来  $m$  行，每行两个数，表示环上的点的坐标。我们定义左上角的点的坐标为  $(0,0)$ ，右下角的为  $(n,n)$ 。要求连续两个点，包括最后一个点和第一个点也相邻。

**Example**

loopy*.in	loopy*.out
5	34
-1 -1 -1 -1 -1	0 0
2 2 -1 -1 3	0 1
2 -1 -1 0 3	1 1
2 -1 -1 2 3	1 2
3 -1 2 2 -1	2 2
	2 3
	1 3
	0 3
	0 4
	0 5
	1 5
	1 4
	2 4
	2 5
	3 5
	3 4
	4 4
	4 5
	5 5
	5 4
	5 3
	4 3
	3 3
	3 2
	4 2
	5 2
	5 1
	5 0
	4 0
	4 1
	3 1
	2 1
	2 0
	1 0

样例解释：



## Scoring

如果得到了正确解，你将得到 10 分。

如果你满足了其中 95% 的限制，你将得到 7 分。

如果你满足了其中 80% 的限制，你将得到 5 分。

如果你满足了其中 70% 的限制，你将得到 4 分。

如果你满足了其中 60% 的限制，你将得到 3 分。

如果你满足了其中 40% 的限制，你将得到 2 分。

如果你满足了其中 20% 的限制，你将得到 1 分。

这里的限制指网格中非 -1 的数字。

如果输出格式不满足条件或者由于自己的原因使得 SPJ 出错将得不到分数。