

Problem A. 猫猫猫猫猫猫虫

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

ID	abbr.	Description	meme
1	/button	bugcat-pressing-the-button	
2	/dead	bugcat-dead	
3	/eat	bugcat-eating	
4	/fn	bugcat-angry	
5	/shui	bugcat-sleeping	
6	/veg	broccolibugcat	
7	/you	fishbugcat	

如图所示，在猫猫虫的国度有一款猫猫虫聊天软件，里面有 7 种表情包。每个表情包均有 3 个属性：

- 编号（ID）：一个 1 至 7 的整数。
- 缩写（abbr.）：由 ‘/’ 开头的，除 ‘/’ 外仅含小写字母的字符串。
- 描述（description）：由小写字母、减号 ‘-’ 组成的字符串。

现在一只猫猫虫给你发了一个字符串，作为该软件的维护人员，你需要判断这个字符串是缩写还是描述，以及其对应的编号是什么。

注意，猫猫虫发送的字符串一定是表格中 7 个表情包对应的缩写或描述。

Input

输入一行，表示猫猫虫发送的字符串。

保证字符串是以下字符串中的一个: `/button`, `/dead`, `/eat`, `/fn`, `/shui`, `/veg`, `/you`, `bugcat-pressing-the-button`, `bugcat-dead`, `bugcat-eating`, `bugcat-angry`, `bugcat-sleeping`, `broccolibugcat`, `fishbugcat`。

Output

输出一行，先是一个字符串 `abbr.` 或 `description`（全小写）分别表示该字符串是缩写、描述。一个空格后一个 1 至 7 的整数表示编号。

Examples

standard input	standard output
<code>/button</code>	<code>abbr. 1</code>
<code>bugcat-angry</code>	<code>description 4</code>

Problem B. 猫猫虫报数

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

现在有一只猫猫虫，它想玩一个报数游戏。它会在一开始限定一个数字 k 。然后从 1 开始报数，它只会报 k 的倍数或十进制位中包含 k 的数。它想知道他报到的第 x 个数是多少。

猫猫虫比较笨，只会一个个数过去，但是这太慢了，于是他把题目交给了你。

Input

两个整数，分别为 k, x ($1 \leq k \leq 9, x \leq 10^5$)。

Output

一个数，代表答案。

Examples

standard input	standard output
3 5	13
4 5	16

Note

对于第一个样例，猫猫虫报到的数是 3, 6, 9, 12, 13..., 其中第 5 个数是 13。

对于第二个样例，猫猫虫报到的数是 4, 8, 12, 14, 16..., 其中第 5 个数是 16。

Problem C. 猫猫虫的队列

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

猫猫虫现在在操作一个队列。猫猫虫有个奇怪的癖好：它不希望自己的队列中出现重复的数字。同时队列中可能出现数字的值被他限定在了 $[1, m]$ 之间。

它总共会进行 Q 次操作，操作共有 3 种：

操作 1：尝试加入一个数 x ，如果 x 已在队列内则不加入，否则加入到队尾。

操作 2：弹出队首。保证队列不为空。

操作 3：查询队列中的第 x 个数，如果不存在第 x 个数的话输出 -1 。

请你对于所有操作 3 输出查询结果。

Input

第一行两个数，代表操作次数 Q 和队列中的数的值域 m ($1 \leq Q \leq 10^6, 1 \leq m \leq 10^6$)。

接下来的 Q 行，首先会输入一个数 $type$ ($1 \leq type \leq 3$) 代表操作类型。

- 如果 $type = 1$ ，则后面输入一个数 x 代表尝试加入的数。保证 $1 \leq x \leq m$ 。
- 如果 $type = 2$ ，则弹出队首。保证队列不为空。
- 如果 $type = 3$ ，则后面输入一个数 x 代表查询。保证 $1 \leq x \leq Q$ 。

Output

对于每个 $type = 3$ 的查询输出答案。

Example

standard input	standard output
10 10	5
1 3	-1
1 5	3
2	8
3 1	
3 2	
1 3	
1 5	
1 8	
3 2	
3 3	

Problem D. 锁

Input file: standard input

Output file: standard output

Time limit: 1 second

Memory limit: 256 megabytes

有 n 个元素 $a_1, a_2, \dots, a_n$ ，每个元素都有一把锁，初始时所有元素均未上锁。

给定 q 个查询，第 i 个查询为四个整数 (t_s, t_e, l, r) ，其中 $1 \leq l \leq r \leq n$ 。每个查询会在时间轴上执行一次上锁过程和一次解锁过程。

上锁过程：从时间 t_s 开始，在接下来的 $r - l + 1$ 个时刻依次检查元素 $a_l, a_{l+1}, \dots, a_r$ 。具体地，在时刻 $t_s + k$ 检查元素 a_{l+k} ($0 \leq k \leq r - l$)。若该元素当前未上锁，则将其上锁，并标记该锁属于当前查询；若已经被其他查询上锁，则不进行操作。即若访问该锁时，锁已被其它上锁过程上锁，这个这把锁不会发生任何变化。

解锁过程：从时间 t_e 开始，在接下来的 $r - l + 1$ 个时刻依次处理元素 $a_l, a_{l+1}, \dots, a_r$ 。具体地，在时刻 $t_e + k$ 处理元素 a_{l+k} ($0 \leq k \leq r - l$)。若该元素当前的锁是由该查询在上锁过程中加上的，则将其解锁；否则不进行操作。

对于同一时刻在同一元素上发生的所有操作，规定按照编号从小到大的顺序依次执行。

对于每个查询，统计其在上锁过程中**成功上锁的元素数量**。

Input

第一行包含两个整数 n, q ($1 \leq n, q \leq 1000$)。

接下来 q 行，每行包含四个整数 t_s, t_e, l, r ($1 \leq l \leq r \leq n, 1 \leq t_s < t_e \leq 10^9$)，表示一个查询。

Output

输出 q 行，第 i 行输出第 i 个查询成功上锁的元素数量。

Examples

standard input	standard output
5 5 3 5 3 4 2 5 4 5 2 4 3 5 2 5 3 5 1 3 5 5	0 1 2 0 1
5 5 1 2 5 5 2 4 3 5 3 4 1 4 1 3 3 5 2 5 1 2	1 0 2 3 2

Problem E. 猫猫虫聚会

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

现在有两只猫猫虫想要聚会。它们生活的地图中有 n 个点，一只猫猫虫会从 1 号点出发，另外一只从 n 号点出发。这张图中共有 m 条边，第 i 条边从 x_i 号点连向 y_i 号点，徒步走的话会消耗 z_i 体力。一次打车需要消耗 1 块钱，可以选择一条边 i ，直接从 x_i 到 y_i 并且不消耗体力。两只猫猫虫在任意时间任意时刻均可打车。

两只猫猫虫的预算总共有 w 块钱。猫猫虫想知道，在预算之内合计最少需要消耗多少体力。如果两只猫猫虫无法相遇的话，输出 -1 。

Input

本题采用多组测试数据。

第一行输入一个数 T ，代表数据组数。

接下来对于每组数据，第一行输入三个数 n, m, w ($\sum n, \sum m \leq 1000, 0 \leq w \leq 1000$)，代表点数，边数和预算钱数。

接下来的 m 行，第 i 行输入三个数 x_i, y_i, z_i ($1 \leq x_i, y_i \leq n, 1 \leq z_i \leq 10^9$) 表示一条从 x_i 到 y_i ，消耗体力为 z_i 的边。

Output

一行，表示答案。如果两只猫猫虫无法相遇的话，输出 -1 。

Example

standard input	standard output
2	1
5 5 2	-1
1 2 3	
1 3 4	
5 4 2	
4 3 1	
2 3 100	
5 4 2	
1 2 3	
1 3 4	
5 4 2	
2 3 100	

Problem F. 卡牌游戏

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 256 megabytes

小 Z 正在玩一款卡牌游戏。他手中有 N 张牌，所有牌正面朝上叠成一摞。每张牌的正面印有一个大写字母。

游戏规则要求他将这摞牌重新整理成新的一摞牌，使得整理后的新牌摞从上到下形成的字母序列的字典序尽可能小。

小 Z 每次只能从这摞牌的最顶端或最底端抽出一张牌，将其放到新牌摞的底部。重复这一操作，直到所有牌都转移到新牌摞为止。

小 Z 提前知道了初始牌摞从上到下的字母序列。请你帮小 Z 算出，按照这种操作方式能得到的字典序最小的牌序。

Input

第一行一个整数 N ($1 \leq N \leq 5 \times 10^5$)。

接下来 N 行，每行一个大写字母，从上到下表示初始牌摞中每张牌上的字母。

Output

输出一个长度为 N 的字符串，表示得到的字典序最小的字母序列。

Example

standard input	standard output
6 B C B D C A	ABCBCD

Note

字符串 s 的字典序小于字符串 t ，当且仅当存在某个位置 i ，使得 $s_1 = t_1, s_2 = t_2, \dots, s_{i-1} = t_{i-1}$ ，且 $s_i < t_i$ ；或者 s 是 t 的前缀且 $s \neq t$ 。其中字符的大小按照字母表顺序比较，即 $A < B < \dots < Z$ 。

例如， $AABC < ABCA$ ，因为在第二个字符处 $A < B$ 。

Problem G. 天网防线

Input file: standard input
Output file: standard output
Time limit: 3 seconds
Memory limit: 1024 megabytes

这是一道交互题。

在一座分层防线中，有一个 $n \times m$ 的网格。行从上到下编号为 1 到 n ，列从左到右编号为 1 到 m 。其中有些格子初始时已经放置了障碍物。

一个入侵无人机会从第 n 行的某个空格出发，试图一路向上突破防线。你需要通过不断增设障碍物，将它拦截在到达第 1 行之前。

设无人机当前位于格子 (r, c) 。若 $r > 1$ ，则它在这一回合会向上移动一格，目标只能是以下三个位置之一： $(r-1, c-1)$ 、 $(r-1, c)$ 、 $(r-1, c+1)$ 。移动后的格子必须满足：

- 仍在棋盘范围内；
- 该格子上没有障碍物。

如果无人机到达了第 1 行，则视为它成功突破防线，你的程序失败。

在无人机每次移动之后，若它当前位于 (r, c) 且 $r > 1$ ，你可以选择放置至多一个新障碍物。若你选择放置，则这个障碍物必须放在某个当前没有障碍物的格子 (x, y) 上，并满足 $x \in \{r-1, r-2\}$ 、 $1 \leq x \leq n$ 、 $1 \leq y \leq m$ 。也就是说，你只能在无人机当前所在位置的上方第 1 行或第 2 行新增障碍。新放置的障碍物会永久保留；你也可以选择这一回合不放置障碍物。

你的目标是让无人机在某个时刻无法继续合法移动，从而成功拦截它。

保证对于所有测试数据，都至少存在一种策略，使你一定能够在无人机到达第 1 行之前将其拦截。

Input

第一行包含两个整数 n 和 m ($2 \leq n, m \leq 100$)，表示棋盘的行数和列数。

接下来 n 行，每行一个长度为 m 的字符串 s_i ，描述初始棋盘状态：

- 若 $s_{i,j} = \#$ ，表示格子 (i, j) 上初始就有障碍物；
- 若 $s_{i,j} = .$ ，表示格子 (i, j) 初始为空。

保证第 n 行至少存在一个空格。

最后一行包含两个整数 r_0 和 c_0 ，表示无人机的初始位置为 (r_0, c_0) 。保证 $r_0 = n$ ，且该位置初始为空。

读取完输入后，按照下述规则进行交互。

Output

Interaction Protocol

读取完输入后，你需要不断进行如下交互，直到交互结束：

1. 输出两个整数 x y 。

- 若 $(x, y) = (0, 0)$ ，表示这一轮你选择不放置障碍物；
- 否则，表示你在格子 (x, y) 放置一个新的障碍物，此时必须满足：
 - 该格子在棋盘范围内；
 - 该格子当前没有障碍物；
 - 若当前无人机位于第 t 行，则必须保证 $x \in \{t-1, t-2\}$ 。

2. 交互器随后让无人机移动一步，并输出两个整数 r 和 c ：

- 若 $(r, c) = (0, 0)$ ，表示无人机已经无法继续进行合法移动，你成功拦截了它。此时你的程序应立即终止；
- 若 $(r, c) = (-1, -1)$ ，表示你拦截失败。出现这种情况时，说明无人机已经到达第 1 行，或者你的输出不合法。此时你的程序应立即终止，你将被判为 **Wrong answer**；
- 否则，表示无人机已经移动到了格子 (r, c) ，随后进入下一轮交互。

若你的程序在读到 $(0, 0)$ 或 $(-1, -1)$ 后仍继续交互，或者没有遵守交互协议，也会得到 **Wrong answer**。

输出后请及时刷新缓冲区，否则可能会得到 **Idleness limit exceeded**。例如：

- C/C++: `fflush(stdout)` 或 `cout.flush()`
- Java: `System.out.flush()`
- Python: `sys.stdout.flush()`

Example

standard input	standard output
5 5	
.....	
.....	
#. #..	
..#..	
.....	
5 3	
	4 4
4 2	
	3 2
0 0	

Note

交互器可以是自适应的。也就是说，无人机每一步移动不一定在交互开始前完全固定。

初始障碍物和你后来放置的障碍物都会一直保留，不会消失。

无人机永远不会进入有障碍物的格子。

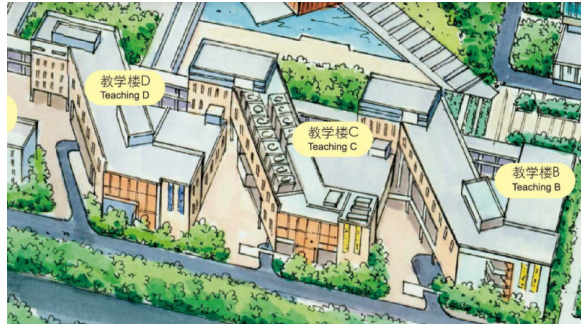
你需要保证：无论交互器如何进行合法移动，你的策略都能最终将其拦截。

上面的样例交互过程如下：

- 初始时无人机位于 (5, 3)。
- 第 1 轮中，你输出 4 4，表示在 (4, 4) 放置一个障碍。此时第 4 行的 (4, 3) 与 (4, 4) 都有障碍，因此无人机只能移动到 (4, 2)。随后交互器输出 4 2。
- 第 2 轮中，你输出 3 2，表示在 (3, 2) 放置一个障碍。此时第 3 行的 (3, 1)、(3, 2)、(3, 3) 都无法到达。随后交互器输出 0 0，表示你已经成功拦截无人机。

Problem H. 设计教学楼

Input file: standard input
 Output file: standard output
 Time limit: 2 seconds
 Memory limit: 1024 megabytes



作为一名香港中文大学（深圳）的学生，小 B 在第一次进入教学楼时迷路了（因为 TD 楼上是 TC，TC 楼上是 TB）。多年以后，小 B 成为了新校区教学楼的设计人，而现在小 B 正在设计教学楼的切面图。

新教学楼将有 $2n$ 个教学区，编号为 $1 \sim 2n$ 。而设计的占地长度为 $2n$ 单位，楼高限为 $n+1$ 单位，这意味着切面图可被视作一个 $(n+1) \times (2n)$ 的网格。而每个格点必须对应一个 $0 \sim 2n$ 的数字， $1 \sim 2n$ 表示该位置的教学区编号，0 表示该位置不属于任何教学区。新教学楼每个教学区在切面图上必须**四连通** *。

若将网格第 i 行、第 j 列的格点的教学区编号表示为 $a_{i,j}$ ，定义教学区 x 在教学区 y ($x \neq y$) 的楼上当且仅当：存在 i, j ($1 \leq i \leq n, 1 \leq j \leq 2n$) 使得 $a_{i,j} = x, a_{i+1,j} = y$ 。

为了结构的条理与优美，小 B 决定了 $(2n-1)$ 条关系，第 i 条关系如教学区 x_i 在教学区 y_i 楼上。若将 x_i 向 y_i 连边，那么 $2n$ 个教学区恰好可以构成一棵以教学区 1 为根的下向树 **。

现在，你需要设计出一个合法的切面图（即确定每个格子的教学区编号），使得这张切面图对应的有向图与小 B 给定的树完全相同。（必须满足所有小 B 给定的关系，必须不满足所有小 B 未给定的关系）

请输出任意一种合法的填充方案，或报告无解。

* **四连通**：对于一个由若干格点组成的集合，如果其中任意两个格点都可以通过一系列相邻（上下左右）的格点（这些格点也属于该集合）连通，则称该集合是四连通的。

** **下向树**：一棵以某个节点为根的有根树，其中每条边的方向都是从父节点指向子节点（即由上层指向下层）。在本题中，给定的 $2n-1$ 条关系教学区 x_i 在教学区 y_i 楼上，对应有向边 $x_i \rightarrow y_i$ ，这些边构成一棵以 1 为根的下向树。

Input

第一行一个整数 T ($1 \leq T \leq 500$)，表示有 T 组数据。

每组数据第一行一个整数 n ($1 \leq n \leq 500$)。

接下来 $2n-1$ 行，每行两个整数 x_i, y_i ，表示一条关系。保证输入构成一棵以 1 为根的下向树。

保证 $\sum n \leq 2000$ 。

Output

对于每组数据：

- 若存在合法方案，输出 $(n + 1)$ 行，每行 $2n$ 个整数表示构造方案。如果有多种方案，输出任意一种即可。
- 若不存在合法方案，输出一行一个字符串 `No solution`。

Example

standard input	standard output
1	2 1 1 1
2	2 2 2 3
1 2	4 4 0 3
1 3	
2 4	

Problem I. 猫猫虫冒险

Input file: standard input

Output file: standard output

Time limit: 3 seconds

Memory limit: 256 megabytes

猫猫虫打开了一款小游戏。这款小游戏中会给你一张无向图，每个点上都有一个怪物，怪物可能会有零个或者一个随从。只有击败这个点所有的敌人（包括随从），才能解锁该点。你能走到某个点当且仅当至少有一个被解锁的点和其相连。

但是走到这个点的时候不一定需要击败所有敌人。也就是说，你可以只击败怪物的随从，随后离开前往其它点。此时这个点并不会解锁。

玩家初始会有一个血量 x ，当尝试杀死一个生命值为 y 的敌人时，如果 $x \geq y$ ，那么敌人死亡，并且玩家的血量加 y 。反之玩家直接死亡。

猫猫虫开始向你求助。它会给你小游戏中的无向图和每个点初始的怪物血量（每个点上有且仅有一个怪物，一开始所有怪物都没有随从）。多次操作，每次询问有两种方式。第一种询问会使点 x 上的怪物召唤血量为 y 的随从（保证每个点 x 上的怪物至多只会召唤一个随从）。第二种询问查询以血量 y 在点 x 开始进行游戏，最终血量的最大值。

Input

第一行三个数 n, m, q ($1 \leq n, m, q \leq 2 \times 10^5$)，代表点数，边数和询问数。

第二行 n 个数，代表每个点的怪物血量 h_i ($1 \leq h_i \leq 10^6$)。

接下来的 m 行，每行 2 个数，代表有条 x 和 y 之间的无向边。保证图中不会有重边和自环。

再接下来的 q 行，第一个数 op 代表询问类型。

如果 $op = 1$ ，那么接下来输入两个数 x, y ($1 \leq x \leq n, 1 \leq y < h_x$)，代表点 x 上会召唤血量为 y 的随从。保证 $y < h_x$ 并且每个 x 最多出现 1 次。

如果 $op = 2$ ，那么接下来输入两个数 x, y ($1 \leq x \leq n, 1 \leq y \leq 10^6$)，代表查询以 y 血量在点 x 开始游戏，最终血量的最大值。

Output

对于每个 $op = 2$ 的询问，输出答案。

Example

standard input	standard output
5 5 5	5
1 2 6 6 8	27
1 2	4
1 3	
1 4	
4 5	
3 5	
2 2 2	
1 3 2	
2 2 2	
1 4 5	
2 4 4	

Problem J. 猫猫虫打麻将

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 1024 megabytes

猫猫虫过年的时候在玩一种以麻将为主题的类小丑牌游戏，游戏中它是自己摸自己打，并且知道牌山中一定的牌。

它突然很好奇，假设自己制作一副有 n 类牌的麻将，然后先摸上初始的 13 张牌后从头开始摸牌。假设自己知晓所有牌山并采取最优策略，那么期望总共需要摸多少张牌（包括初始的 13 张）才能自摸四暗刻？答案对 $10^9 + 7$ 取模。

注意，这里认为每张麻将牌互不相同。

麻将定义： n 类牌，每一类有 4 张。因此总共有 $4n$ 张麻将牌。

四暗刻定义：我们将相同的 3 张牌定义为一个暗刻。四暗刻胡牌需要四个暗刻和一个对子。

更加清晰的题意：

现在有 $4 \times n$ 张牌，总共被染成了 n 种颜色，其中每种颜色四张牌。相同颜色的牌也视作互不相同的牌。

接下来将其随机打乱，共有 $(4n)!$ 种打乱方式。定义一种打乱方式的代价为最小的 i 满足前 i 张牌中有 4 种颜色出现过 3 次及以上并有 5 种颜色出现 2 次及以上。求打乱后的代价期望。答案对 $10^9 + 7$ 取模。

Input

一个数，表示 n ($1 \leq n \leq 2 \times 10^6$)。

Output

一个数，代表期望对 $10^9 + 7$ 取模的答案。

Examples

standard input	standard output
5	80495372
4	0

Problem K. 落月摇情满江树

Input file: standard input

Output file: standard output

Time limit: 3 seconds

Memory limit: 1024 megabytes

江边的树可以被描绘为 x 个编号为 $1 \sim x$ 结点的形式:

- 有且仅有一个根, 编号为 1。
- 编号为 i ($i \neq 1$) 的结点恰好有一个父亲, 编号为 f , 满足 $f < i$ 。

小 B 无法记住”树”这样抽象的东西, 于是发明了两种将树变为排列的方式:

Algorithm 1 Transfer Tree to Permutation

Require: sons of each node *son*, interger *type* equals to 1 or 2.

Ensure: queue *q*.

```

1: function DFS(u)
2:   q.push(u)
3:   if type = 1 then
4:     for each v in son[u] in increasing order do
5:       DFS(v)
6:     end for
7:   else
8:     for each v in son[u] in decreasing order do
9:       DFS(v)
10:    end for
11:  end if
12: end function
13: DFS(1)

```

- 方式一, $type = 1$: 根开始, 先访问父亲, 再访问儿子, 儿子按编号从小到大访问。
- 方式二, $type = 2$: 根开始, 先访问父亲, 再访问儿子, 儿子按编号从大到小访问。

可以证明访问完所有节点后得到的 q 中的元素是 $1 \sim x$ 的一个排列。

而一个排列是有歧义的当且仅当它既可以通过一棵树 t_1 由方式一得到, 也可以通过一棵树 t_2 由方式二得到。注意两棵树可以完全相同, 即 t_1 可以等于 t_2 。

猫猫虫只记住了一个长为 $n + m$ 的排列 a 中 $1 \sim n$ 的出现位置, 并让你推测对于可能的所有排列, 有多少种排列是有歧义的。

Input

第一行一个整数 T ($1 \leq T \leq 10^5$) 表示数据组数。

每组数据第一行两个整数 n 和 m ($0 \leq m \leq 80, 1 \leq (n+m) \leq 10^5$)。ac 第二行 $n+m$ 个整数 $a_{1 \sim n+m}$ ($0 \leq a_i \leq n$)，若 $a_i = 0$ ，表示第 i 个位置猫猫虫没有记住， a 中 $1 \sim n$ 的每个数恰好出现一次。。
保证 $\sum(n+m) \leq 3 \times 10^5$ 。

Output

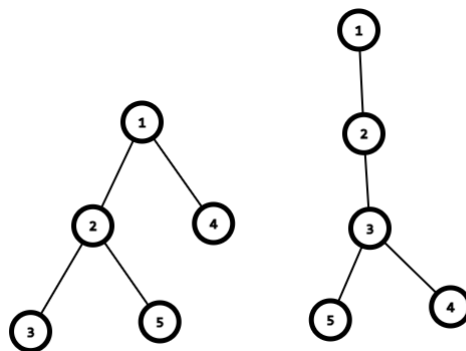
对于每组数据输出一行一个整数表示有歧义的排列的数量，对 $10^9 + 7$ 取模。

Example

standard input	standard output
3	1
3 0	1
1 2 3	109
4 1	
1 2 3 0 4	
3 5	
1 2 3 0 0 0 0 0	

Note

对于第二组数据，可能的排列只有 1,2,3,5,4 一种。可通过图一和方式一得到，也可通过图二和方式二得到，因此有歧义。



Problem L. 宴会餐厅

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 1024 megabytes

有 2^n 只猫猫虫，属性两两不同，且可以表示为一个 $0 \sim 2^n - 1$ 的数字。

现要将这些猫猫虫两只一组分为 2^{n-1} 组去餐厅就餐，猫猫虫心地善良，不会有任何一只同时在两组内。

餐厅有编号 $1 \sim 2^{n-1} + 1$ 共 $(2^{n-1} + 1)$ 张二人桌子，要求每桌的两只猫猫虫属性值异或和为桌子编号。

但是此时先来餐厅独自就餐的小 B 已经找到了全餐厅最安静的 x 号桌就坐。（即编号为 x 的桌子不能再坐猫猫虫）

你是否能提供一种分组，使得所有不同组的猫猫虫均可就坐于不同的二人桌，且小 B 不移动位置？

Input

输入一行两个整数 n, x ($1 \leq n \leq 20, 1 \leq x \leq 2^{n-1} + 1$)。

Output

输出第一行一个字符串 YES 或 NO 表示是否存在一种分组。

若存在，接下来 2^{n-1} 行每行两个整数 x_i, y_i ，表示属性为 x_i 和属性为 y_i 的猫猫虫在一组内。

可按任意顺序输出每组猫猫虫，若有多种分组方法，输出任意一种即可。

Examples

standard input	standard output
3 1	YES 5 7 0 3 2 6 1 4
2 1	NO
3 2	NO
1 2	YES 0 1