

Problem A. 题目背景是 GPT 给的

Input file: standard input
Output file: standard output

GPT 编的题意

在遥远的符文王国，流传着一块神秘的魔法玉盘。玉盘上刻有 n 处符文，每一处都蕴藏着不同的元素之力，记录在一个数组 A 中。传说，当王国陷入混乱，只有将这 n 处符文的力量完全统一，才能唤醒盘中的终极力量，拯救苍生。

此处省略两万字。

简要题意

给定一个长度为 n 的数组 $A = [A_0, A_1, \dots, A_{n-1}]$ 。定义对数组的"第 i 轮操作"为：对于所有 $0 \leq j < n$ ，同时将

$$A_j \leftarrow A_j \mid A_{(i+j) \bmod n},$$

其中 \mid 表示按位或（bitwise OR）运算。

任务：求最小的整数 T ，使得经过 T 轮上述操作后，数组 A 中的所有元素都相同。

Input

第一行一个整数 T ($1 \leq T \leq 5$)—表示数据组数，每组数据的输入格式如下：

每组数据包含两行，第一行一个整数 n ($2 \leq n \leq 10^5$)，表示数组长度。

第二行包含 n 个整数 A_0, A_1, \dots, A_{n-1} ($0 \leq A_i \leq 10^9$)。

Output

每组数据输出仅包含一行一个整数，表示答案。

Example

standard input	standard output
3	1
3	0
1 2 3	2
3	
1 1 1	
7	
7 6 5 4 3 2 1	

Problem B. 列车

Input file: standard input
Output file: standard output

在某国有一条铁路线，总共有 n 个站点。有 m 辆列车，编号 $1 \sim m$ ，其中编号 i 的列车容量 c_i 人，并从 l_i 站发车依次经过 $l_i + 1, l_i + 2, \dots, r_i - 1, r_i$ 站。所有乘客必须在 l_i 上车，但可以在途中提前下车。
现在有 10^{114514} 个人处于 1 号站点，他们都想去往 n 号站点。在合理购票的情况下，最多有多少人能到 n 号站点？

Input

第一行包含一个整数 T ($1 \leq T \leq 1000$)—表示数据组数，每组数据的输入格式如下：
第一行包含两个整数 n, m ($2 \leq n \leq 10^9, 1 \leq m \leq 2 \times 10^5$)—表示站点数量和列车数量。
接下来 m 行，其中第 i 行包含三个整数 l_i, r_i, c_i ($1 \leq l_i \leq r_i \leq n, 1 \leq c_i \leq 10^9$)—表示起点站、终点站和列车容量。
保证所有数据组中 m 的和不超过 2×10^5 。

Output

对于每组数据输出一个整数表示最多能有多少人到达 n 号站点。

Example

standard input	standard output
2	5
3 1	3
1 3 5	
10 2	
1 6 4	
5 10 3	

Problem C. 区域划分

Input file: standard input
Output file: standard output

你有一张 $n \times n$ 的方格图，你需要将这 $n \times n$ 个格子划分为 n 个区域，每个格子恰好属于一个区域，每个区域至少有一个格子，区域编号 $1 \sim n$ ，并且每个区域的格子不一定是连通的。

完成划分后，你需要用最少的颜色对每个格子进行涂色，但需要满足以下要求：

- 相同区域的格子的颜色必须相同。
- 相邻区域的格子颜色必须不同。（区域 i 与区域 j 相邻，当且仅当存在一个区域 i 中的格子与一个区域 j 中的格子共享一条边）

此时，使用颜色的数量称为这个划分的着色数。你需要找到一种划分区域的方式，使得这种划分的着色数最大。

如果有多种方法使得着色数最大，则输出任意一种都算正确。

Input

第一行包含一个整数 T ($1 \leq T \leq 100$)—表示数据组数，每组数据输入格式如下：

每组数据仅一行一个整数 n ($1 \leq n \leq 1000$)—表示方格图大小/需要划分的区域个数。

保证所有组数据中 n^2 的和不超过 10^6 。

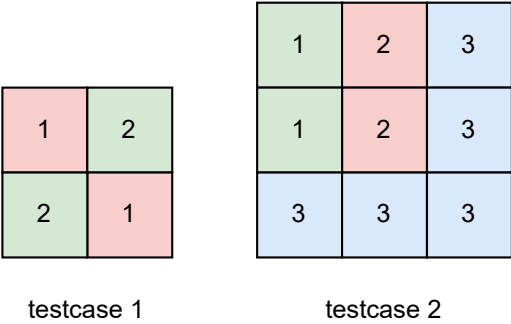
Output

对于每组数据，输出 n 行，每行 n 个整数，第 i 行第 j 个整数表示方格图中第 i 行第 j 个格子划分到的区域编号。

Example

standard input	standard output
2	1 2
2	2 1
3	1 2 3
	1 2 3
	3 3 3

Note



Problem D. 黄金替罪羊

Input file: standard input
Output file: standard output

小 W 有一个长度为 $2n$ (n 为偶数) 的指令序列 S ，其中包含 ‘L’，‘R’ 两种指令，其中 ‘L’ 表示往数轴负方向走 1 个单位，‘R’ 表示往数轴正方向走 1 个单位。

小 W 和小 B 会依据指令序列 S 在数轴上进行一个有趣的游戏，该游戏的具体流程如下：

- 初始时，小 W 出现在数轴原点。
- 接下来 n 步，小 W 会按照 S_1, S_2, \dots, S_n 的指令内容依次行动。
- n 步后，小 B 会出现在原点（注意此时若小 W 也在原点也认为二者在此时相遇），接下来第 $n+1$ 步至 $2n$ 步，第 $n+k$ ($1 \leq k \leq n$) 步时小 W 会按照 S_{n+k} 的指令内容行动一步，同时小 B 会按照 S_k 的指令内容行动一步。若在第 $n+k$ ($k = 0, 1, \dots, n$) 步行动后，小 B 和小 W 出现在同一位置，则认为二者在此时相遇。
- 共 $2n$ 步执行完毕后，游戏结束。

若依据该指令序列 S 执行完整个游戏后，小 W 和小 B 从未相遇，则称该指令序列 S 是合法的。

现在，你得到了一个长度 $2n$ 的序列 T ，其除了 ‘L’，‘R’ 两种字符之外，还包含 ‘?’ 字符。你需要将 T 中每个 ‘?’ 字符替换为 ‘L’ 或者 ‘R’，由此得到一个指令序列 S 。你需要求出所有能得到的合法指令序列 S 的数目，由于方案数可能很大，答案请对 998244353 取模。

Input

输入第一行包含一个正整数 n ($1 \leq n \leq 50$) 且 n 为偶数。

输入第二行包含一个长度为 $2n$ 的字符串序列 T ，包含 ‘L’，‘R’，‘?’ 三种字符，含义如题目描述所示。

Output

输出一行一个整数，表示所有合法的指令序列数目对 998244353 取模的结果。

Example

standard input	standard output
4 L?L?L??R	3

Note

对于样例中的情况，能得到的合法指令序列数目包括 LLLLLLLR，LRLLLLLR 和 LLLRLLLLR 共 3 种情况，故输出 3。

Problem E. 异或问题

Input file: standard input
Output file: standard output

给定五个非负整数 n, l, x, y, k ，你需要找到连续的 l 个非负整数 $a, a + 1, a + 2, \dots, a + l - 1$ ，要求这些数都要小于等于 n 并且对每个数按位异或 x 后（即 $a \oplus x, (a + 1) \oplus x, (a + 2) \oplus x, \dots, (a + l - 1) \oplus x$ ），这些数中小于等于 y 的数字个数恰好有 k 个。

Input

第一行包含一个正整数 T ($1 \leq T \leq 2 \times 10^4$)—表示数据组数，每组数据的输入个数如下：
仅一行包含 5 个非负整数 n, l, x, y, k ($0 \leq n, x, y \leq 10^9, 1 \leq k \leq l \leq n + 1$)—意义如题所述。

Output

一个非负整数 a 表示找到的答案。若有多个答案，任意输出一个即可；若不存在答案，输出 -1 。

Example

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

Note

第一组数据中，选择 $a = 2$ ， $\{2, 3, 4\}$ 异或 $x = 5$ 后变为 $\{7, 6, 1\}$ ，其中小于等于 $y = 4$ 的恰好有 $k = 1$ 个。
第二组数据中，选择 $a = 3$ ， $\{3, 4, 5\}$ 异或 $x = 5$ 后变为 $\{6, 1, 0\}$ ，其中小于等于 $y = 4$ 的恰好有 $k = 2$ 个。
第三组数据中，无论怎么选，都不能使得恰好有 $k = 3$ 个数异或后小于等于 $y = 4$ 。

Problem F. 赢麻了

Input file: standard input
Output file: standard output

Alice 和 Bob 在玩决斗游戏，决斗游戏只比拼战斗力的大小。

- 第一天，Alice 拥有 6 的战斗力，Bob 拥有 5 的战斗力，Alice 比 Bob 战斗力高。所以是 Alice 赢了，Alice 会说 “Win”。
- 第二天，Alice 拥有 6 的战斗力，Bob 拥有 500 的战斗力，Alice 比 Bob 战斗力低。所以 Alice 没有赢，但是 Alice 有个战斗力为 1000 的朋友 Candy。Alice 认为，虽然 Bob 赢过了自己，但是没有赢过 Candy，等于 Bob 输了。Bob 输了，也就是 Alice 赢了。这时 Alice 会说 “WIN”。
- 第三天，Alice 拥有 6 的战斗力，Bob 拥有 5000 的战斗力，Alice 比 Bob 战斗力低，Alice 的朋友 Candy 的战斗力 1000，也没有 Bob 高。Alice 怎么都赢不了。但是赢不了有个赢字，也等于赢了。这时 Alice 会说 “nowin”。

给定三个整数 a, b, c ，分别表示 Alice，Bob 和 Candy 的战斗力。

- 如果是第一天的情况，也就是 $a > b$ ，输出 “Win”。
- 否则，如果是第二天的情况，即 $c > b$ ，输出 “WIN”。
- 其他情况，输出 “nowin”。

Input

第一行一个整数 T ($1 \leq T \leq 100$)—表示数据组数，每组数据的输入格式如下：
每组数据一行，包含 3 个整数 a, b, c ($0 \leq a \leq 10^9, 0 \leq b \leq 10^9, 0 \leq c \leq 10^9$)。

Output

每组数据输出仅包含一行，表示答案。注意答案区分大小写。

Example

standard input	standard output
5	Win
6 5 1000	WIN
6 500 1000	nowin
6 5000 1000	WIN
1234 4009 4039	nowin
6 6 6	

Problem G. LCA & MST

Input file: standard input
Output file: standard output

给定一棵有 n 个节点的有根树 T ，树的根为节点 1。每个节点 i 有一个初始权值 w_i 。
接下来，根据树结构 T 和当前的权值数组 $w = [w_1, w_2, \dots, w_n]$ 构造如下图 $G = (V, E)$:

- 1. G 是一个完全图，顶点集 $V = \{1, 2, \dots, n\}$;
- 2. 每一对不同的节点 (u, v) ($u, v = 1, 2, \dots, n$ 且 $u \neq v$) 之间有一条边，其边权为 $w_{\text{LCA}(u,v)}$ ，即 u 和 v 在根为 1 的树 T 中的最近公共祖先的编号所对应的权值。

定义 $L_T(w)$ 为图 G 的最小生成树（MST）的边权之和。
接下来有 q 次操作。每次操作给出两个正整数 p 和 c ，表示:

- 1. 找出树中以节点 p 为根的子树（包含 p 自身）中的所有节点组成的集合 S_p ;
- 2. 对所有 $x \in S_p$ 执行操作: $w_x := w_x + c$;

你需要在第一次操作前和每次操作后，回答当前 $L_T(w)$ 的值。

Input

第一行两个整数 n, q ($2 \leq n \leq 2 \times 10^5, 1 \leq q \leq 2 \times 10^5$)，分别表示树的节点数和操作数。
接下来 $n - 1$ 行，每行包含两个整数 u, v ($1 \leq u, v \leq n$ 且 $u \neq v$)，表示树 T 中一条无向边 (u, v) 。
接下来一行包含 n 个整数 w_1, w_2, \dots, w_n ($1 \leq w_i \leq 10^9$)，表示初始权值。
接下来 q 行，每行两个整数 p 和 c ($1 \leq p \leq n, 1 \leq c \leq 10^4$)，表示一次子树操作。

Output

输出 $q + 1$ 行，每行一个整数，表示第一次操作前及每次操作后，函数 $L_T(w)$ 的值。

Example

standard input	standard output
7 4	34
1 2	34
1 3	38
2 4	68
2 5	72
4 6	
4 7	
7 6 5 4 3 2 1	
5 4	
4 3	
1 5	
2 2	

Problem H. 连接

Input file: standard input
Output file: standard output

在二维平面上有一个矩形方框，在方框的下边界排列着写着 $1 \sim n$ 数字（每个数字恰好一个）的 n 个小球，他们都紧贴着下边界。在方框的上边界同样排列着写着 $1 \sim n$ 数字（每个数字恰好一个）的 n 个小球，但只有一部分紧贴上边界。现在总共有 n 个可以弯曲变形的绳子，你需要将写着相同数字的两个小球通过绳子连接在一起，但任意两个绳子不能交叉，且绳子不能从紧贴上边界的小球的上方绕过，也不能从紧贴下边界的小球的下方绕过。你需要回答是否可以做到。

Input

第一行包含一个整数 T ($1 \leq T \leq 10^3$)—表示数据组数，每组数据的输入格式如下：
第一行包含一个整数 n ($1 \leq n \leq 2 \times 10^5$)—表示上边界和下边界的小球个数。
第二行包含 n 个整数 a_1, a_2, \dots, a_n ($1 \leq a_i \leq n$)—表示上边界的小球上写的数字。
第三行包含 n 个整数 b_1, b_2, \dots, b_n ($1 \leq b_i \leq n$)—表示下边界的小球上写的数字。
第四行包含 n 个整数 c_1, c_2, \dots, c_n ($c_i \in \{0, 1\}$)— $c_i = 0$ 表示上边界的第 i 个小球没有紧贴上边界， $c_i = 1$ 表示上边界的第 i 个小球紧贴上边界。
保证 a, b 是中 $1 \sim n$ 的每个数字出现恰好一次，保证所有组数据的 n 之和不超过 2×10^5 。

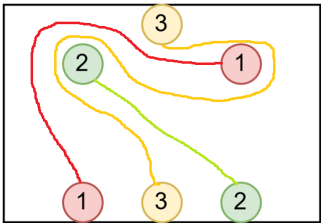
Output

对于每组数据，输出 “Yes” 表示可以用 n 个不交叉的绳子将标有相同数字的小球连接，输出 “No” 表示不行。

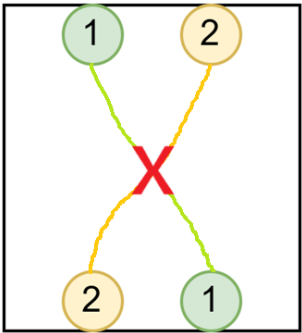
Example

standard input	standard output
2	Yes
3	No
2 3 1	
1 3 2	
0 1 0	
2	
1 2	
2 1	
1 1	

Note



(a) 第一组数据



(b) 第二组数据

Problem 1. 答辩

Input file: standard input
Output file: standard output

在课程期末答辩环节，老师要求 n 名同学依次上台进行答辩。每名同学答辩结束后，需要至少回答其他同学提出的 k 个问题。同时为了防止有同学不积极提问，还要求每名同学至少提出 k 个问题。
此外，老师把他们分为 m 组，每位同学恰好属于一个组，并提出了以下额外的要求：

- 1. 任何人不能向同组的同学提问。
- 2. 如果 x 向 y 提了问，那么 y 就不能向 x 提问。
- 3. 任何人最多只能向同一个人提问一次。

你需要判断是否存在一种提问方案，使得每名同学至少回答了 k 个问题，至少提出了 k 个问题，并且满足老师额外的要求。
可能有多种不同的提问方案满足条件，你只需要输出任意一种满足条件的即可，如果不存在一种方案，输出 “No”。

Input

第一行包含一个正整数 T ($1 \leq T \leq 100$)—表示数据组数，每组数据的格式如下：
第一行包含三个正整数 n, m, k ($3 \leq n \leq 1000, 2 \leq m \leq n, 1 \leq k \leq \lfloor \frac{n-1}{2} \rfloor$) — 分别表示人数、组数和提问/回答的个数。
接下来共有 m 行，第 i 行的开头是一个正整数 b_i ($1 \leq b_i \leq n$) — 表示第 i 组的人数，接下来包含 b_i 个正整数 $a_{i,1}, a_{i,2}, \dots, a_{i,b_i}$ ($1 \leq a_{i,j} \leq n$) — 表示这一组的同学的编号。
保证所有组数据中 n^2 之和不超过 10^6 。

Output

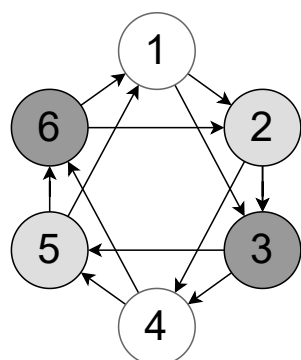
对于每组数据，输出格式如下：
如果不存在一种方案，输出 “No”。
否则，在第一行输出 “Yes”，接下来 n 行，其中第 i 行开头一个正整数 c_i — 表示第 i 个同学提的问题数量，接下来 c_i 个正整数，表示每个问题提问的对象。
若有多种方案，任意输出一种即可。

Example

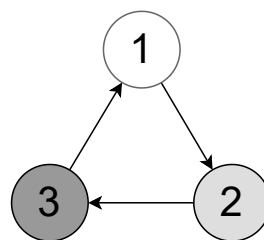
standard input	standard output
3	Yes
6 3 2	2 2 3
2 1 4	2 3 4
2 2 5	2 4 5
2 3 6	2 5 6
3 3 1	2 6 1
1 1	2 1 2
1 2	Yes
1 3	1 2
5 2 2	1 3
1 1	1 1
4 2 3 4 5	No

Note

前两组数据构造如图所示，颜色深浅表示不同组，箭头表示提问的方向。



testcase 1



testcase 2

Problem J. RGB 树

Input file: standard input
Output file: standard output

小 W 有一棵包含 n 个节点的无根树 T ，每个节点初始颜色为红色（R）、绿色（G）或蓝色（B）之一。

现在，小 W 可以选择 T 上若干节点将其染为白色（W）。节点被染为白色后，它的颜色将不再被视为红色、绿色或蓝色。

小 W 的目标是，通过将一些节点染为白色，使得染色后下列三个条件同时成立：

- 对于所有节点对 (u, v) ，若 u 和 v 的颜色均不是红色，则它们在树中的简单路径上不包含任何红色节点；
- 对于所有节点对 (u, v) ，若 u 和 v 的颜色均不是绿色，则它们在树中的简单路径上不包含任何绿色节点；
- 对于所有节点对 (u, v) ，若 u 和 v 的颜色均不是蓝色，则它们在树中的简单路径上不包含任何蓝色节点。

其中，节点对 (u, v) 满足 $u \neq v$ 且 $u, v \in \{1, 2, \dots, n\}$ ；简单路径是指一条不包含重复节点的路径。

请你求出，为了满足上述条件，最少需要将多少个节点染为白色？

Input

输入第一行包含一个正整数 n ($2 \leq n \leq 2 \times 10^5$)，表示树的节点数。

输入第二行包含长度为 n 的字符串 S ，其中第 i 个字符为 S_i ($S_i \in \{\text{'R'}, \text{'G'}, \text{'B'}\}$)，表示节点 i 的初始颜色。

接下来的 $n - 1$ 行，每行两个整数 u, v ($1 \leq u, v \leq n$ 且 $u \neq v$)，表示树中的一条无向边 (u, v) 。

Output

输出一行一个整数，表示最少需要染为白色的节点个数。

Example

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

Note

对于样例中的情况，小 W 可以选择将节点 2 和 3 染为白色，此时要求的三个条件同时成立。

可以证明，这是能使得染为白色的节点数目最小的唯一方案。

Problem K. 消消乐

Input file: standard input
Output file: standard output

小 W 有一个 $n \times n$ 的方格图，初始时每个格子中数字均为 1，且有 $2n$ 个正整数 a_1, a_2, \dots, a_n 和 b_1, b_2, \dots, b_n 。

小 W 会在这个方格图上玩一个名为"消消乐"的游戏。初始时，设集合 $S = T = \{1, 2, \dots, n\}$ 。接下来会发生若干次操作，每次操作，小 W 会从 $|S| + |T|$ 种可能事件中随机选择一个，具体概率分布如下：

- 对于每个 $i \in S$ 有 $\frac{a_i}{\sum_{i \in S} a_i + \sum_{j \in T} b_j}$ 的概率将第 i 行全标为 0，随后将整数 i 从集合 S 中删除；
- 对于每个 $j \in T$ 有 $\frac{b_j}{\sum_{i \in S} a_i + \sum_{j \in T} b_j}$ 的概率将第 j 列全标为 0，随后将整数 j 从集合 T 中删除。

本次操作的代价为**操作完后**整个方格图中每个格子的数字之和。

共 $2n$ 次操作后，此时恰有 $S = T = \emptyset$ ，游戏结束。

请你帮小 W 计算，整个游戏中的期望代价之和为多少？答案对 998244353 取模。

Input

第一行包含一个整数 n ($1 \leq n \leq 10^5$) 表示网格的大小为 $n \times n$ 。

第二行包含 n 个正整数， a_1, a_2, \dots, a_n ($1 \leq a_i \leq 5 \times 10^4$) 含义如题目描述所示。

第三行包含 n 个正整数， b_1, b_2, \dots, b_n ($1 \leq b_i \leq 5 \times 10^4$) 含义如题目描述所示。

Output

输出一行一个整数，表示所求答案对 998244353 取模的结果。

Example

standard input	standard output
2 1 1 1 1	665496238

Problem L. 栈与重复

Input file: standard input
Output file: standard output

给你一个栈 S ，初始为空，接下来有 n 个操作，每个操作是如下操作之一：

- **Push x** ：将非负整数 x 压入栈 S 。
- **Pop**：将 S 中栈顶元素弹出，**题目保证执行该语句时栈 S 始终非空**。
- **Repeat**：重复执行一遍此前的所有语句（不包含本语句）。具体地，若这是操作序列中的第 i 个操作，则依次执行前 $i - 1$ 个操作（包括此前的 **Repeat** 操作）。

请求出每次操作后 S 中所有数字的对 998244353 的取模结果。

Input

第一行一个整数 n ($1 \leq n \leq 2 \times 10^5$)，表示操作次数。

接下来 n 行，每一行包含一个操作，操作的输入格式见题目描述。保证 **Push** 操作中输入的非负整数 x 满足 $0 \leq x < 998244353$ 。

Output

输出共 n 行，其中第 i 行输出第 i 次操作执行结束后，栈 S 中所有元素之和对 998244353 取模的结果。

Example

standard input	standard output
6	1
Push 1	2
Repeat	1
Pop	3
Push 2	6
Repeat	4
Pop	

Note

对于样例中的情况，初始时栈 $S = []$ ，接下来 6 次操作变化情况如下：

1. 执行操作 **Push 1**，此时栈 S 变为 $[1]$ ， S 中数字之和为 1。
2. 执行操作 **Repeat**，即重复一遍 **Push 1**，此时栈 S 变为 $[1, 1]$ ， S 中数字之和为 2。
3. 执行操作 **Pop**，此时栈 S 变为 $[1]$ ， S 中数字之和为 1。
4. 执行操作 **Push 2**，此时栈 S 变为 $[1, 2]$ ， S 中数字之和为 3。
5. 执行操作 **Repeat**，则需重复执行一遍此前的所有语句，此时栈 S 变为 $[1, 2, 1, 2]$ ， S 中数字之和为 6。
6. 执行操作 **Pop**，此时栈 S 变为 $[1, 2, 1]$ ， S 中数字之和为 4。