Problem A. 极巨团体战

时间限制: 1 second

在极巨团体战中，n 名玩家需要每人派出一只宝可梦参战，齐心协力捕捉极巨化宝可梦。假定这些玩家只能选择喵头目或苍响中的一只参战。

每只喵头目的攻击力为 100，每只苍响的攻击力为 200。每只喵头目可以使所有 n 只宝可梦的攻击力乘以 1.1。多只喵头目的效果可以叠加，也就是说，如果你派出了 t 只喵头目，攻击力会乘以 1.1^t。

请你求出 n 只宝可梦的攻击力之和最大可以是多少。

输入格式

一行包含一个整数 n $(1 \le n \le 50)$，含义见题目描述。

输出格式

一行一个实数，表示答案。若输出的值与标准答案的绝对误差或相对误差在 10^{-4} 之内，则认定输出正确。具体地说，假设你给出的答案为 a，而标准答案为 b，则你的答案被认为正确，当且仅当 $\frac{|a-b|}{\max(1,|b|)} \le 10^{-4}$。

样例

standard input	standard output
2	400.000000000000000
10	2593.742460100002063

提示

对于第一个样例，派出 2 只苍响是一种最优解。

对于第二个样例，派出 10 只喵头目是一种最优解。

本题的数值不是真实数值，请宝可梦玩家注意。

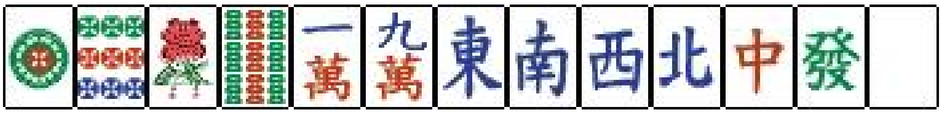
Problem B. 国士无双

时间限制: 1 second

日本麻将共有 34 种牌。在游戏进行时，每名玩家拥有 13 张牌，称作“手牌”。玩家按照顺序从“山牌”中摸 1 张牌，然后打出 1 张牌。

玩家若希望通过**自摸**的方式“和”牌，则需要使自己的 13 张牌与自己刚从“山牌”中摸到的 1 张牌，组成特定的某个牌型。

国士无双是日本麻将中的一种特殊的和牌牌型，牌型为指定的 13 种牌各 1 张，再额外加上这 13 种中的某 1 张。



国士无双所指定的那 13 种牌

四个自摸“和”国士无双的情况，最右侧的牌为刚从“山牌”中摸到的牌

小 A 现在在打单机麻将，但是由于打不过电脑，他打算通过**自摸**的方式“和”国士无双。

当前是小 A 的回合，并已经摸到 1 张牌还没打出去，即共有 14 张牌。我们用 a_i ($1 \le a_i \le 34$, $1 \le i \le 14$) 来表示小 A 现在的第 i 张牌。

若 $1 \le a_i \le 13$，则表示小 A 的第 i 张牌是国士无双牌型中所指定的某一种牌；若 $14 \le a_i \le 34$，则表示小 A 的第 i 张牌**不在**国士无双的牌型中。

当小 A 共有 $1, 2, \ldots, 13$ 中各 1 张，并另外还有 1 到 13 中的某 1 张，则认为小 A 成功自摸“和”国士无双。

假设每种牌都有无限张，请问小 A **最少**再经过几个回合才能自摸“和”国士无双呢?

形式化地说，给定一个长度为 14 的序列 $\{a_i\}$，每次可以修改任意一个元素成为任意一个数，那么最少多少次修改才能让序列只包含 $1, 2, \ldots, 13$ 中的数，且 $1, 2, \ldots, 13$ 各有至少一个?

输入格式

一行共 14 个正整数 a_i ($1 \le a_i \le 34$)，表示小 A 当前的手牌。

数据并不保证这副牌，在真实麻将中合法。

输出格式

一个整数，表示自摸“和”国士无双所需要的最少的回合数。如果当前已经组成了国士无双的和牌牌型，则输出 0。

样例

standard input	standard output
1 2 3 4 5 6 7 8 9 10 11 12 13 14	1
1 1 1 1 2 2 2 2 3 3 3 3 4 4	9

提示

注意，在此题中我们不考虑任何其他的和牌方式与和牌形式。（对于打过麻将的选手：和平常不一样的是，只允许自摸，只允许和这一种和牌型，同时山牌和手牌不一定是真实麻将的情况。）

第一个样例中，我们需要将 14 打出，接下来若摸到 $1 \sim 13$ 中的任何一张即可自摸“和”国士无双。

第二个样例中，我们可以依次打出 $1, 1, 1, 2, 2, 2, 3, 3, 3$，并能相应地依次摸到 $5, 6, 7, 8, 9, 10, 11, 12, 13$，就可以自摸和国士无双。可以证明这是达到用最少的回合数自摸“和”国士无双的方案之一。

Problem C. 亦或骗子

时间限制: 2 seconds

对于非负整数 u, v 的异或 $w = u \oplus v$，w 二进制下第 i 位为 1，当且仅当 u 和 v 二进制下的第 i 位不相同。

非负整数 x 的 $\text{popcount}(x)$，定义为 x 二进制下 1 的个数，比如 $\text{popcount}(128) = 1$, $\text{popcount}(26) = 3$, $\text{popcount}(255) = 8$。

现给一个长度为 n 的数组 $a_1, a_2, \ldots, a_n$。定义数组连续的一段 $a_l, a_{l+1}, \ldots, a_r$ 的**分数**为该段内所有数异或的 popcount，即：

$$\text{popcount}\left(a_l \oplus a_{l+1} \oplus \cdots \oplus a_r\right)$$

需要请你给出如下信息：

1. 请给出数组 a_i 的一个分段方案，使得各段**分数**的总和**最大**；
2. 请给出数组 a_i 的一个分段方案，使得各段**分数**的总和**最小**。

输入格式

第一行，一个正整数 n $(1 \le n \le 10^5)$，表示数组的长度。

第二行，共 n 个非负整数 $a_1, a_2, \ldots, a_n$ $(0 \le a_i < 2^{20})$，表示题目描述中的数组 a_i。

输出格式

按顺序输出两个数组分段的方案，第一个方案需使得分数和**最大**，第二个方案需使得分数和**最小**。

方案的格式为：

- 一行，n 个正整数 s_i $(1 \le s_i \le n, 1 \le i \le n)$，表示原数组中第 i 个整数分在了第 s_i 段。需要满足 $s_1 = 1$，以及 $s_{i+1} \in \{s_i, s_i + 1\}$

注意：方案不需要段数最小或最大，只要各段分数总和最大化或最小化即可。如果有多种方案，则给出任意一种即可。

样例

standard input	standard output
6 6 4 2 4 1 7	1 2 3 3 3 4 1 1 1 2 2 2
9 3 9 1 10 9 6 7 1 6	1 2 3 3 4 4 5 6 6 1 1 1 1 1 2 2 2 3
3 0 0 0	1 1 1 1 2 3

提示

第一个样例中，最大的各段分数总和为 9，最小的各段分数总和为 1。

第二个样例中，最大的各段分数总和为 17，最小的各段分数总和为 3。

第三个样例中，最大的各段分数总和为 0，最小的各段分数总和也为 0。

Problem D. 树上路径

时间限制: 1 second

小 A 是一个普通的初中生。

小 A 所在学校的校门前有着一排 n 棵树，从西向东依次编号为 $1, 2, \ldots, n$。相邻两棵树间的距离都是 1 个单位。

小 A 上课的教学楼恰好在树 1 旁，所以每个课间，小 A 都会走出教室，上树活动。下课后，小 A 会从树 1 出发，一路从西向东跳到树 n，总共恰好经过 m 棵树。

由于小 A 睡眠很充足，因此她每次跳跃会移动**至少** k 个单位的距离。

我们把小 A 从树 1 跳到树 n 经过的 m 棵树（包含树 1 和树 n）的编号的集合 S，叫做一条树上路径。

小 A 喜欢各种不同的树上路径，现在她想知道，符合以上描述的不同的树上路径一共有多少种。小 A 认为两条树上路径不同，当且仅当其中一条路径中存在至少一个编号，满足该编号不存在于另一条路径中。

由于答案可能很大，你只需要判断答案是否为奇数。

输入格式

第一行一个正整数 T $(1 \le T \le 10^5)$，表示数据组数。

接下来 T 行，每行表示一组数据，包括一行三个正整数 n, m, k $(2 \le n, m \le 10^9,\ 1 \le k < n)$，分别表示树的数量、树上路径的大小和每次跳跃移动的最小距离。

输入保证存在至少一条满足条件的、从树 1 跳到树 n 的树上路径。

输出格式

输出 T 行，每行一个字符串 `yes` 或 `no`。第 i 行输出表示第 i 组数据中，不同的树上路径的数量是否为奇数。如果是奇数，输出 `yes`，否则输出 `no`，大小写不敏感。

样例

standard input	standard output
4 10 3 2 5 3 1 15 5 3 29 4 5	no yEs nO Yes

提示

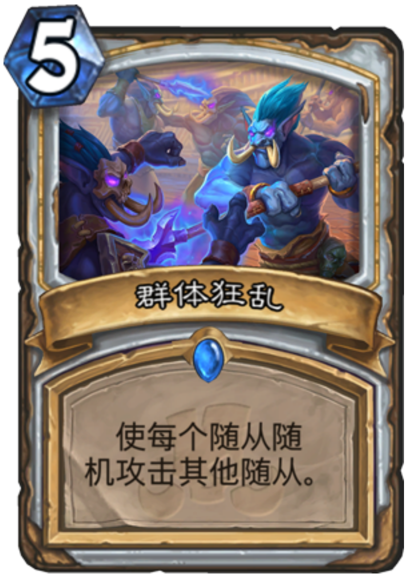
样例中，第一组数据中树上路径有 6 条，分别为：$\{1, 3, 10\}$, $\{1, 4, 10\}$, $\{1, 5, 10\}$, $\{1, 6, 10\}$, $\{1, 7, 10\}$, $\{1, 8, 10\}$。

样例中，第二组数据中树上路径有 3 条，分别为：$\{1, 2, 5\}$, $\{1, 3, 5\}$, $\{1, 4, 5\}$。

Problem E. 群体狂乱

时间限制: 1 second

在炉石传说游戏里，有一张牌叫群体狂乱。



它的效果是：使每个随从随机攻击其它随从。

具体来说，我们首先均匀地随机随从的攻击顺序，然后随从按照顺序进行攻击。如果当前随从存活，那么均匀地随机选取一个除当前随从以外的存活的随从，进行攻击；但如果没有其余存活的随从，则跳过，不进行攻击。如果当前随从已经不再存活，则同样跳过，不进行攻击。

现有 n 个随从在场上，第 i 个随从拥有攻击力 atk_i 和血量 hp_i。我们不考虑随从的各种特效，每次攻击时，参与攻击的双方的血量各自减少对方的攻击力。当某个随从的血量小于或等于 0 时，则认为该随从不再存活。

例如：一个 5 攻 2 血的随从攻击一个 1 攻 11 血的随从，之后 5 攻 2 血的随从会变成 5 攻 1 血，而 1 攻 11 血的随从则变成了 1 攻 6 血。

请问一次群体狂乱后，每个随从的存活概率。

输入格式

第一行，一个正整数 n $(1 \le n \le 6)$，表示随从数。

接下来共 n 行，每行包含两个整数 atk_i, hp_i $(0 \le atk_i \le 10^6, 1 \le hp_i \le 10^6)$，分别为第 i 个随从的攻击力和血量。

输出格式

一共输出 n 行，每一行一个小数，第 i 行输出表示第 i 个随从存活的概率。

若所有输出的值与标准答案的绝对误差或相对误差在 10^{-6} 之内，则认定输出正确。具体地说，假设对于某个输出的值，你给出的答案为 a，而标准答案为 b，则你的答案被认为正确，当且仅当 $\frac{|a-b|}{\max(1,b)} \le 10^{-6}$。

样例

standard input	standard output
3 1 1 1 1 1 1	0.333333333333 0.333333333333 0.333333333333
2 2 6 3 3	0.000000000000 0.000000000000
2 1 7 3 3	1.000000000000 1.000000000000
3 0 1 0 2 1 1	0.208333333333 0.583333333333 1.000000000000

提示

即使随从只有 0 点攻击力，它也会主动去攻击一个目标，但是它不能对对方造成伤害，只会挨打。

一个死去的随从主动尝试攻击会被跳过；场上只有一个随从时攻击会被跳过。

在第一个样例中，第一个进行攻击的随从会与随机另一个互相消灭，之后剩下的两次攻击均被跳过。所以每一个随从都有 $\frac{1}{3}$ 的概率存活。

在第二个样例中，一种情况是第一个随从先攻击第二个随从，然后第二个随从攻击第一个随从，双方都不再存活；另一种情况是第二个先攻击第一个、第一个再攻击第二个，这样两个随从也都无法存活。

在第三个样例中，无论谁先攻击，最后两者都剩余 1 血，即均存活。

Problem F. woafrnraetns 与正整数

时间限制: 1.5 seconds

woafrnraetns 喜欢正整数。他收集了一个由 n 个正整数组成的序列 $a_1, a_2, \ldots, a_n$，以及另外两个正整数 p 和 q $(p < q)$。

某天，woafrnraetns 开始研究这个正整数序列，他希望找到序列中的两个数 a_i, a_j $(1 \le i < j \le n)$，满足 $\frac{a_i}{a_j}$ 在 $\left[\frac{p}{q}, \frac{q}{p}\right]$ 之间，即 $\frac{p}{q} \le \frac{a_i}{a_j} \le \frac{q}{p}$。请你帮他解决这个问题。

本题数据规模较大，因此采用了较为特别的输入方式。具体来说，我们输入序列的前 m $(1 \le m \le \min(n, 3 \times 10^5))$ 项，后面的 $n - m$ 项则采用一定的规则生成。若 $m < n$，那么生成规则为：$a_{i+1} = ((b \cdot a_i + c) \bmod t) + 1$ $(m \le i < n)$，其中 b, c, t 在输入中给出，$z \bmod t$ 为 z 除以 t 的最小非负余数。

输入格式

第一行包含三个整数 n, p, q $(2 \le n \le 3 \times 10^7, 1 \le p < q \le 10^4)$，含义见题目描述。

第二行包含四个整数 m, b, c, t $(1 \le m \le \min(n, 3 \times 10^5), 0 \le b, c < t \le 10^9)$，含义见题目描述。

第三行包含 m 个整数，第 i 个为 a_i $(1 \le a_i \le t)$。

输出格式

如果无解，输出一行一个整数 -1。

如果存在解，则输出一行两个整数 i, j，表示 a_i, a_j 满足题目要求，两个整数以一个空格分隔。i, j **应当满足** $1 \le i < j \le n$。

如果有多种答案，输出任意一个即可。

样例

standard input	standard output
3 1 2 3 0 0 100 1 2 5	1 2
2 1 2 2 0 0 100 1 3	-1
3 2 5 1 2 1 998244353 1	2 3

提示

对于第一个样例，正整数序列为 $1, 2, 5$，其中 $\frac{1}{2} \le \frac{1}{2} \le \frac{2}{1}$。

对于第二个样例，正整数序列为 $1, 3$。由于 $\frac{1}{3} < \frac{1}{2}$，因此不存在满足要求的答案。

对于第三个样例，正整数序列为 $1, 4, 10$，其中 $\frac{2}{5} \le \frac{4}{10} \le \frac{5}{2}$。

Problem G. 林克与宝箱咒语

时间限制: 1 second

织梦岛是由 n 个地区组成的小岛，地区之间由 $n-1$ 条道路连接。任意两个地区都可以直接或者间接地通过道路连通，也就是说，这 n 个地区形成了一棵树。每条道路上都有一个宝箱，每个宝箱有一个种类，用 1 到 n 之间的整数来标记。

现在林克要从地区 u 出发，沿着最短的路径走到地区 v。在行走的过程中，林克可以开启他遇到的宝箱，当然也可以选择跳过遇到的宝箱。由于林克需要沿最短路径行走，当然是不能返回去开之前跳过的宝箱的。如果到达终点 v 后，林克开启的宝箱种类序列与某个古老的咒语相同，那么他将能获得一份神秘的大礼。

现在，林克还没有确定旅途的起点和终点，因此他希望你能帮他统计所有方案中有机会获得大礼的数量。具体地说，你需要统计满足下列要求的**有序对** (u,v) 的数量：林克从地区 u 出发，沿最短路径走到地区 v 结束，他在路上能够合理地开启或跳过宝箱，使得开启的宝箱种类序列与古老的咒语相同。林克事先知道整个地区的地图以及每个宝箱的种类。

注意：$u \neq v$ 时，(u,v) 和 (v,u) 是两个**不同**的有序对。

输入格式

第一行包含两个整数 n,k $(1 \le n \le 10^5, 1 \le k \le 200)$。$n$ 表示地区的数量，k 表示古老咒语的长度。

接下来 $n-1$ 行，每行包含三个整数 u,v,w $(1 \le u,v \le n, u \neq v, 1 \le w \le n)$。表示地区 u 和 v 之间有一条道路，道路上的宝箱种类为 w。保证给出的 n 个地区形成了一棵树。

接下来一行包含 k 个整数 x_i $(1 \le x_i \le n)$。第 i 个整数 x_i 表示古老咒语的第 i 项。

输出格式

输出一行包含一个整数，表示答案，即问题中有序对的数量。

样例

<table>
<tr><th>standard input</th><th>standard output</th></tr>
<tr><td>3 1
1 2 1
1 3 1
1</td><td>6</td></tr>
<tr><td>4 2
1 2 1
1 3 2
1 4 1
1 2</td><td>2</td></tr>
<tr><td>1 1
1</td><td>0</td></tr>
</table>

提示

对于第一个样例，所有的 $(u,v)(u \neq v)$ 均满足要求。

对于第二个样例，满足要求的有序对是 $(2,3)$ 和 $(4,3)$。

Problem H. 宝可梦与分支进化

时间限制: 1 second

在宝可梦世界中，进化是一种十分神奇的现象。宝可梦在积累了足够的经验后，不仅外形会发生显著的变化，各项能力也会得到飞跃般的提升，这种现象就被称为宝可梦的进化。而更为神奇的是，有些宝可梦根据外部环境的不同，可以进化成多种不同的形态，这被称为宝可梦的分支进化。

现在大木博士正在研究一个宝可梦家族，这个家族中共有 n 种宝可梦，我们用 $1, 2, \ldots, n$ 给它们编号。其中，宝可梦 1 是初始的宝可梦，宝可梦 i 则由宝可梦 $f_i (1 \le f_i < i, i \ge 2)$ 进化而来。

大木博士的院子里有 m 只宝可梦，它们从左向右依次排好了队，左起第 i 只宝可梦的种类为 a_i。大木博士希望从中选取一个宝可梦**子序列**，使得除第一只宝可梦外，每只宝可梦都能由前一只宝可梦进化一次或多次得到。为了便于研究，他希望选取的宝可梦数量最多。

形式化地说，大木博士希望选取一些下标 $i_1, i_2, \ldots, i_k$，满足 $1 \le i_1 < i_2 < \cdots < i_k \le n$，且宝可梦 $a_{i_{j+1}}$ 可以由宝可梦 a_{i_j} 一次或多次进化得到（$1 \le j < k$），并且 k 最大。

请你求出大木博士最多能选取多少只宝可梦。

输入格式

第一行包含两个整数 n, m，n 表示宝可梦的种类数，m 表示大木博士院子里的宝可梦数量。保证 $2 \le n \le 5 \times 10^5, 1 \le m \le 5 \times 10^5$。

接下来一行包含 $n-1$ 个整数，第 i 个整数为 f_{i+1}，表示宝可梦 $i+1$ 由宝可梦 f_{i+1} 进化而来。保证 $1 \le f_{i+1} < i+1$。

接下来一行包含 m 个整数，第 i 个整数为 a_i，表示左起第 i 只宝可梦的种类。保证 $1 \le a_i \le n$。

输出格式

输出一行包含一个整数，表示大木博士至多可以选取多少只宝可梦。

样例

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

提示

对于第一个样例，最优的子序列是 $1, 2, 4$。

对于第二个样例，整个序列都可以选取。

Problem I. Poison AND^OR Affection

时间限制: 1 second

毒牙か慈愛か、それとも…

Poison AND÷OR Affection

小钾是一名音乐游戏玩家！现在，他正在回顾几年前的一场比赛记录——「G2R2014 "GO BACK 2 YOUR ROOTS"」。在这场比赛中，三名作曲者需要组成一个队伍进行创作，同时每人提交一份作品，共计三个作品参赛。在结束投稿后的展览期间，观众可以自由游玩所有队伍的作品，并为喜欢的作品提交评分。队伍所有作品获得的总分越高，队伍的排名也越靠前。

展览阶段持续了 k 天。一个作品在展览期间共获得了 n 个评分，且按照评分时间**升序**排序后依次为 $a_1, a_2, \ldots, a_n$。作品在某一天获得的所有评分是序列中连续的一段 $a_l, a_{l+1}, \ldots, a_r$，且当天的影响力值可由下式计算：

$$f(l, r) = (a_l \wedge a_{l+1} \wedge \cdots \wedge a_r) \oplus (a_l \vee a_{l+1} \vee \cdots \vee a_r)$$

其中 $\wedge$ 为按位与，$\vee$ 为按位或，$\oplus$ 为按位异或。一个作品获得的总影响力值，是它在每一天的影响力值的总和。

小钾找到了某个作品在 k 天内的评分，且按照评分时间**升序**排序后依次为 $a_1, a_2, \ldots, a_n$。然而由于数据丢失，他并不知道每一个评分具体是在第几天给出的。现在，小钾想知道，在依次为所有评分重新指定一个日期，且每天至少有一个评分后，作品的总影响力值最大可能是多少。请注意日期的指定应当合法：对于 $a_1, a_2, \ldots, a_n$，它们对应的评分日期非降。

请你编写一个程序，帮小钾解决这个问题。

输入格式

第一行，两个空格分开的整数 n, k $(1 \le k \le n \le 2,000)$，含义见题目描述。

第二行，共 n 个整数 a_i $(0 \le a_i \le 10^9)$，代表作品按照评分时间**升序**排序后的评分。

输出格式

输出一行一个整数，表示作品可能达到的最大总影响力值。

样例

standard input	standard output
5 2 3 1 2 5 4	9
7 4 11 45 14 19 19 8 10	94

提示

样例 1 中，一种可能的指定日期为：

- 第 $1,2$ 个评分在第 1 天给出，获得的评分为 $f(1,2)=(3\wedge 1)\oplus(3\vee 1)=2$。
- 第 $3,4,5$ 个评分在第 2 天给出，获得的评分为 $f(3,5)=(2\wedge 5\wedge 4)\oplus(2\vee 5\vee 4)=7$。

作品的总影响力值为 $f(1,2)+f(3,5)=2+7=9$。

在 C, C++, Python 3 和 Java 中，按位与、按位或、按位异或这三项操作分别对应了 “&”、“|”、“^” 三个运算符（不含引号）。含义分别为，两个非负整数进行相应操作后，结果中二进制某一位上为 1，当且仅当操作数的同一位上 “均为 1”、“某一个为 1”、“恰好只有一个为 1”。

Problem J. 括号序列

时间限制: 1 second

合法的括号序列定义如下：

- “()” 是一个合法的括号序列。
- 若 “A” 是一个合法的括号序列，则 “(A)” 也是一个合法的括号序列。
- 若 “A”, “B” 是一个合法的括号序列，则 “AB” 也是一个合法的括号序列。
- 比如，“((())())” 是一个合法的括号序列，“)()(” 不是一个合法的括号序列。

现给出一个长度为 n 的合法的括号序列 S，以及 m 个区间括号变更操作。

区间 l,r 的括号变更操作，会将当前维护的括号序列中，第 l 到 r 个括号进行修改，将这一段中的左括号 “(” 更改为右括号 “)”，同时右括号 “)” 更改为左括号 “(”。

操作以最初给定的括号序列作为当前维护的括号开始进行，按顺序对当前维护的括号序列进行修改。

请在每次操作之后，判断修改后的序列是否是合法的括号序列。

输入格式

第一行，两个整数 n,m $(2 \le n \le 10^5, 1 \le m \le 10^5)$，**保证 n 为偶数**，分别表示括号序列的长度和操作的数量。

第二行，一个长度为 n 的**合法的**括号序列 S。

接下来 m 行，每行两个整数 l_j, r_j $(1 \le l_j \le r_j \le n)$，其中的第 j 行表示第 j 次操作的参数，进行题目所描述的操作。

输出格式

输出 m 行，每行为字符串 `yes` 或 `no`，大小写不敏感。

若第 j 个输出为 `yes`，则表示按顺序进行完第 j 个操作后，括号序列**合法**；若第 j 个输出为 `no`，则表示按顺序进行完第 j 个操作后，括号序列**不合法**。

样例

standard input	standard output
4 8 (()) 2 3 2 3 2 4 2 2 3 4 1 2 3 4 1 4	yeS YeS no no yEs no NO yeS
8 12 ((()())) 4 5 6 7 1 6 2 7 1 2 4 7 5 8 2 6 1 5 1 2 7 8 2 8	yeS No no no yEs yes NO NO No nO YeS No

提示

在第一个样例中，操作后的括号序列如下表：

l_j	r_j	S	is valid?
-	-	(())	-
2	3	()()	yes
2	3	(())	yes
2	4	()((	no
2	2	((((	no
3	4	(())	yes
1	2	))))	no
3	4	))((	no
1	4	(())	yes