

## 题目 A. Another A+B Problem

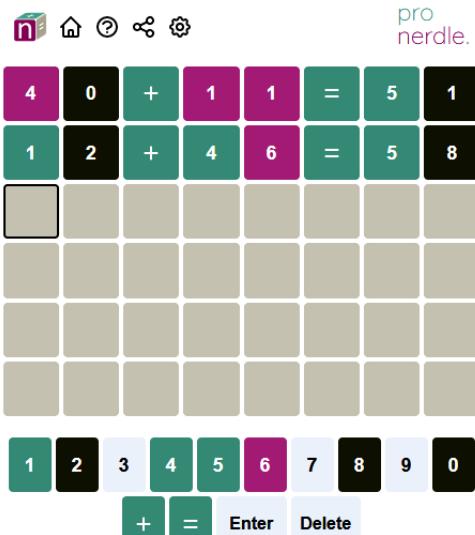
输入文件: 标准输入

输出文件: 标准输出

时间限制: 1 秒

内存限制: 1024 MB

在 2022 年上海市大学生程序设计竞赛之前, 你想玩一些游戏来放松一下。今天, 你的朋友淘米向你介绍了一个叫 Nerdle 的游戏, 主要内容是来猜一些等式。



一局 Nerdle 游戏。

你每次都可以提交一个等式, 交互程序会用格子的颜色来返回信息。

- 如果格子是绿色的, 如图中第一行的  $+$ ,  $=$  和 5, 那么这个数字或运算符就在答案的正确的位置上。
- 如果格子是紫色的, 如图中第一行的 4 和前两个 1, 那么这个数字或运算符在答案中, 但位置错误。
- 如果它是黑色的, 说明该数字或运算符不在答案中, 或
- 如果你的等式中包含了一个比答案中出现次数更多的数字, 你将得到与答案中出现次数相同的非黑格子, 其余为黑色。因此, 图中的第一个等式正好包含两个 1, 但不在这个等式任何一个出现 1 的地方。请注意, 如果对于一个数字来说的紫色和黑色格子都存在, 那么交互程序可以任意选择哪些是黑色的。

玩了一段时间后, 你发现大部分时间浪费在等号前后凑相等上了。作为一个强大的算法竞赛选手, 你觉得这种人类劳动效率太低。你决定用电脑分析某个等式的结果, 筛选出的可能满足条件的等式。该等式和答案有以下限制。

- 第三和第六个字符固定为绿色的 ‘+’ 和 ‘=’。
- 唯一允许的运算符是二元加号。所以, 答案仅可能是 “ $??+??=??$ ” 的形式, 其中 ‘?’ 代表一个数字 ( $0 \dots 9$ )。
- 允许前导零, 即 “ $01+02=03$ ” 满足条件。

请写一个程序, 通过一个给定的等式的结果找出所有可能的合法等式。

## 输入

输入包含两个长度为 8 的字符串  $E, C$ , 分两行, 分别表示所问的等式和交互程序返回的颜色。 $C$  由‘GPB’组成, 分别表示绿色、紫色和黑色。

输入保证:

- $E$  是合法等式。
- $E$  的第三和第六个字符固定为 ‘+’ 和 ‘=’, 且其他位置为数字 (0...9)。
- $C$  的第三和第六个字符固定为 ‘G’.

## 输出

第一行输出一个整数  $R$ , 表示剩余可能的不同等式的数量。如果两个等式至少有一个字符不同, 则视为不同。

在接下来的  $R$  行中, 每行输出一个字符串, 表示与输入对应的合法等式。

你可以按任何顺序输出等式。输入保证至少有一个对应的合法等式存在。

## 样例

标准输入	标准输出
40+11=51 PBGPPGGB	7 11+42=53 11+43=54 11+44=55 11+45=56 11+46=57 11+47=58 11+48=59
12+46=58 GBGGPGGB	6 11+45=56 13+43=56 15+41=56 16+40=56 16+41=57 16+43=59
11+22=33 PBGPPGGP	1 22+13=35
11+22=33 BPGPPGGP	1 22+13=35
01+02=03 PPGPPGPP	2 10+20=30 20+10=30

## 注释

从第一个和第二个样例中可以得出结论, 插图中游戏的唯一解是:  $11+45=56$ 。

第三和第四个样例是本质相同的, 然而黑色和紫色的选择是不同的。

## 题目 B. Bracket Query

输入文件: 标准输入  
输出文件: 标准输出  
时间限制: 2 秒  
内存限制: 1024 MB

括号序列是对称美的典型代表。这里，我们讨论的是由小括号‘(’和‘)’形成的括号序列。例如，“((())”，“()((())”都是合法的括号序列。

形式化地讲，一个括号序列是合法的，当且仅当它可以通过以下规则构造出来：

- “()”是合法的括号序列。
- 如果  $A$  是一个合法的括号序列，那么在前面加一个左括号并在后面加一个右括号形成的  $(A)$  是一个合法的括号序列。
- 如果  $A, B$  是两个合法的括号序列，那么把两个括号序列连接起来形成的  $AB$  是一个合法的括号序列。

今天早柚要和你玩一个游戏。她有一个合法的长度为  $n$  的括号序列  $S$ ，你可以向她提出如下格式的问题：“ $S$  的第  $l$  个字符到第  $r$  个字符形成的子串中，左括号和右括号的数量之差的是多少？”，然后早柚会告诉你答案。

你想使用上述方法来确定这个括号序列。不过，在玩了一会儿游戏后，早柚熟练地施展疾风遁术溜去睡觉了。“大丰收！可以睡觉了吗？”

你需要基于现有的询问和回答来确定一个可能的括号序列。由于早柚没睡醒，可能她的回答存在矛盾，在这种情况下你也需要判断出来。

### 输入

第一行包含两个整数  $n, q$  ( $2 \leq n \leq 3000, 0 \leq q \leq \min(\binom{n+1}{2}, 5 \times 10^5)$ )，分别表示括号序列的长度和现有查询的个数。

接下来  $q$  行中，第  $i$  行包含三个整数  $l_i, r_i, c_i$  ( $1 \leq l_i \leq r_i \leq n, -n \leq c_i \leq n$ )，表示第  $i$  查询范围  $q_i = [l_i, r_i]$ ，以及查询的回答是  $c_i$ ，表示左括号数减去区间内右括号数。

输入保证  $n$  为偶数，且询问区间两两不同。

### 输出

如果没有合法解，输出“?”。

否则，输出一行“!  $S$ ”，其中  $S$  是一个满足输入条件的合法括号序列。

### 样例

标准输入	标准输出
4 1 1 2 0	! ()()
4 1 1 2 2	! (())
2 2 1 1 1 2 2 -1	! ()
2 1 1 1 2	?
4 0	! ()()

## 注释

对于第一组和第二组样例来说，长度为 4 的合法括号序列只有 “((())” 和 “(())”。通过询问前两个字符形成的子串，我们可以依据答案是 2 还是 0 来分辨他们。

## 题目 C. Coffee Overdose

输入文件: 标准输入  
输出文件: 标准输出  
时间限制: 3 秒  
内存限制: 1024 MB

你正在做你的后天就要交的超级大作业。你还有一些时间, 但... 你已经快要睡着了, 你感受到重力牵引着你的脑袋不停往下做功。

别担心! 你至少还有足够的咖啡 – 咖啡含有的咖啡因会让你保持清醒。不过, 你注意到的一个问题: 咖啡会让你在一段时间内保持清醒, 但在这之后, 你会更加疲惫。为了提高工作效率, 你必须明智地使用你的抗疲劳武器。

我们用体力  $S$  来表示你的精神状态。在每一秒钟, 你为你的项目贡献  $S$  的完成度, 之后你的体力会减少 1。当你的体力减少到 0 或更少时, 你会彻底失控并进入梦乡。

你可以在每秒钟刚好开始时喝下一杯咖啡, 效果将会持续  $C$  秒。在咖啡的持续时间里, 你不能再喝另一杯咖啡, 同时你的体力也将被固定在开始喝咖啡时的状态。在效果结束后, 你的体力将立刻减少  $C + 1$ , 也即咖啡会让你额外感到 1 点体力的疲劳。

你的目标是在你睡着前爆发出生产力。对于给定的  $S$  和  $C$ , 你需要给出一个最优的安排, 使总的完成度最大化。

### 输入

第一行包含一个整数  $T$  ( $1 \leq T \leq 10^5$ ), 表示测试组数。

对于每组数据, 单独一行包含两个整数  $S, C$  ( $1 \leq S, C \leq 172\,800 = 2 \times 24 \times 60 \times 60$ ), 表示你的初始耐力和一杯咖啡的持续时间。

### 输出

对于每组数据, 输出一行一个整数, 表示可能的完成度最大值。

### 样例

标准输入	标准输出
4	2
1 2	3
2 1	63
10 4	29859840000
172800 172800	

### 注释

对于第一组数据, 最开始就喝一杯咖啡是最优的, 此时体力序列将会是 1, 1, -2。

对于第二组数据, 适度咖啡益脑, 过度咖啡伤身, 停止过量服用, 享受健康生活。

对于第三组数据, 一种最优的规划是分别在第 3 秒和第 7 秒开始时喝一杯咖啡, 此时体力序列将会是 10, 9, 8, 8, 8, 8, 3, 3, 3, 3, -2。

## 题目 D. Demonstrational Sequences

输入文件: 标准输入  
输出文件: 标准输出  
时间限制: 3 秒  
内存限制: 1024 MB

数学小天才咖波最近学会了怎样计算两个数之间的最大公约数 (Greatest Common Divisor, GCD)。欧几里得算法 (也叫辗转相除法) 是一个历史悠久的计算最大公约数的算法, 这个递归算法对于两个正整数  $x, y$  的最大公约数求解流程如下:

- 如果  $y \neq 0$ , 递归计算  $\text{gcd}(y, x \bmod y)$ ;
- 否则, 返回  $x$ 。

咖波觉得这个算法过于普及了, 因此不太有趣。他觉得更有意思的是, 通过一些无穷数列来展示一些 GCD 特有的无意义性质。现在, 他有两个正整数  $P$  和  $Q$  满足  $Q|P$  ( $a|b$  表示  $a$  是  $b$  的约数, 即  $b \bmod a = 0$ )。此外, 他有  $k$  个正整数对  $\{(a_1, b_1), (a_2, b_2), \dots, (a_k, b_k)\}$ , 他将使用这些数对来生成  $k$  个无穷数列。其中, 第  $i$  个无穷数列  $\{x_{i,0}, x_{i,1}, x_{i,2}, \dots\}$  将通过下列规则生成:

- $x_{i,0} = a_i$
- $x_{i,j} = x_{i,j-1}^2 + b_i$  ( $j > 0$ )

咖波定义一个无穷数列  $\{x_0, x_1, x_2, \dots\}$  是有意义的, 当且仅当存在两个非负整数  $u$  和  $v$  满足  $0 \leq v < u$ , 且  $\text{gcd}(x_u - x_v, P) = Q$ , 这里的  $\text{gcd}(a, b)$  表示  $a$  和  $b$  之间的最大公约数。

对于每个无穷数列, 咖波需要你帮助他判断其是否为有意义的数列。

### 输入

第一行包含三个整数  $P, Q, k$ , ( $1 \leq P \leq 2^{32} - 1$ ,  $1 \leq Q \leq 2^{20}$ ,  $1 \leq k \leq 200$ ), 保证  $Q|P$ 。

此后  $k$  行, 每行包含两个正整数。第  $i$  行的两个正整数  $a_i$  和  $b_i$  ( $1 \leq a_i, b_i \leq 2^{64} - 1$ ), 表示第  $i$  个正整数对。

### 输出

输出一个长度为  $k$  的 0/1 串。如果第  $i$  个无穷数列是有意义的, 则第  $i$  个字符为 1, 否则为 0。

### 样例

标准输入	标准输出
15 5 5 1 1 1 2 2 4 4 8 8 16	11010
998244352 1048576 3 2022 924 12345678 1234567 23333333 6666666	001

### 注释

对于第一个样例:

- 第一个无穷数列  $\{x_{1,0}, x_{1,1}, x_{1,2}, \dots\}$  是  $\{1, 2, 5, 26, \dots\}$ 。考虑  $u = 3, v = 0$  时,  $\gcd(x_u - x_v, P) = \gcd(26 - 1, 15) = 5 = Q$ , 因此第一个数列是有意义的。
- 第二个无穷数列是有意义的, 因为存在  $u = 2, v = 0$  满足条件。
- 第四个无穷数列是有意义的, 因为存在  $u = 2, v = 1$  满足条件。
- 可以证明, 第三个和第五个无穷数列都不是有意义的。

## 题目 E. Expenditure Reduction

输入文件: 标准输入  
输出文件: 标准输出  
时间限制: 1 秒  
内存限制: 1024 MB

经商之道，不外乎开源节流。为了让财务情况更良好，相比开源，节流容易的多。

淳先生是一家会员制餐厅的经理。由于大环境的影响，餐厅无法负担包含过多的新鲜美味菜肴的菜单。淳经理想在保留特色菜的同时，让菜单更精简。

目前的菜单可以被看做一个由小写英文字母与数字构成的字符串  $S$ ，其中淳经理认为菜单的灵魂是  $S$  的一个子序列  $F$ 。缩减菜单时，你可以选择一个  $S$  的子串  $S'$ ，满足  $F$  仍然是  $S'$  的子序列。淳经理希望你从  $S$  中找到满足要求的最短子串  $S'$ 。

这里给出子序列和子串的定义。考虑两个非空字符串  $A, B$ :

- 如果我们说  $A$  是  $B$  的子序列，意思是存在一个大小为  $|A|$  的下标集合  $\{i_k\}$  满足  $1 \leq i_1 < i_2 < \dots < i_{|A|} \leq |B|$ ，且  $A = B_{i_1} B_{i_2} \dots B_{i_{|A|}}$ 。
- 如果我们说  $A$  是  $B$  的子串，意思是你可以从  $B$  中删去一个（可能为空的）前缀和一个（可能为空的）后缀得到  $A$ 。

### 输入

第一行包含一个整数  $T$  ( $1 \leq T \leq 10^4$ )，表示测试组数。

对于每组数据，输入一行用空格隔开的两个字符串  $S, F$  ( $1 \leq |S| \leq 10^5, 1 \leq |F| \leq 100$ )。输入保证  $F$  是  $S$  的子序列，并且两者均由小写英文字母（‘a’到‘z’）和数字（‘0’到‘9’）组成。

保证  $T$  组数据中  $\sum |S|$  不超过  $5 \times 10^5$ 。

### 输出

对于每组数据，在一行中输出一个字符串，表示最短的包含  $F$  作为子序列的  $S$  的子串。

如果有多个答案，输出任何一个即可。

### 样例

标准输入	标准输出
4 114514 15 shanghai cpc ac aaabbbaaabbbccc abc howdeliciousandfreshit is oishii	145 aic abbbc owdeliciousandfreshiti

## 题目 F. Forest of Magic

输入文件: 标准输入  
 输出文件: 标准输出  
 时间限制: 8 秒  
 内存限制: 1024 MB

魔法之森, 顾名思义, 是一片充满魔法的森林。在魔法之森里, 住着一位年轻的魔法使——雾雨魔理沙。

魔法之森中有许多地点, 它们被若干双向道路连接着, 每条道路都连接着一对不同的地点。每一对地点都可以直接或间接地通过道路互相到达, 并且从一个地方到另一个地方只有一条简单路径。换言之, 森林中的地点和道路形成一棵树的结构。

最初, 森林里有  $n$  个地点, 编号从 1 到  $n$ 。魔理沙的家在 1 号地点, 所以从魔理沙的视角来看, 魔法之森可以被视为一棵以 1 号地点为根的树。

众所周知, 许多失落之物会进入幻想乡, 因此魔法之森中时不时会有新的地点出现。每个新出现的地点恰好有一条双向的道路与现有的某个地点相连, 所以每当新的地点出现后, 魔法之森仍然是一棵树。

魔法之森也有很多顽皮的妖精。有时会有一只妖精从一个特定的地点沿道路飞到另一个地点, 由于妖精是大自然的化身, 在她经过的每一个地点, 都会新长出  $k$  朵特定颜色的花。花朵的颜色可以被编号成 1 到  $10^9$  之间的整数。

魔理沙喜欢赏花, 有时她也会邀请朋友一起赏花。在每次邀请朋友之前, 她会选择一个特定的地点, 并统计出在它的子树中, 颜色编号不大于  $c$  的花的数量。这里的子树定义为: 如果  $u$  位于  $v$  和 1 之间的简单路径上, 那么  $v$  就在  $u$  的子树中。注意, 魔理沙不喜欢摘花, 因此这样的操作不会引起任何变化。

然而魔法之森实在是太大了, 因此魔理沙找到了你, 希望你能帮她统计。另外, 由于魔理沙不喜欢等待太长时间, 所以每次她询问时, 你都必须立刻回答她。

### 输入

输入的第一行包含两个整数  $n$  ( $1 \leq n \leq 3 \times 10^4$ ) 和  $q$  ( $0 \leq q \leq 10^5$ ), 表示最开始魔法之森中有  $n$  个地点, 以及魔理沙后面将进行  $q$  次操作。

接下来  $n - 1$  行, 每行包含两个正整数  $u_i$  和  $v_i$  ( $1 \leq u_i, v_i \leq n$ ,  $u_i \neq v_i$ ), 表示有一条双向道路直接连接  $u_i$  和  $v_i$  号地点。

接下来  $q$  行, 每行按顺序表示一个操作, 格式形如以下三种之一:

- 1  $u_i$ : 一个新的地点出现在了魔法之森中。假设此次操作之前已有  $n'$  个地点, 则新的地点的编号为  $n' + 1$ 。另外, 有一条双向道路直接连接新的地点与  $u_i$  ( $1 \leq u_i \leq n'$ ) 号地点。
- 2  $u_i v_i c_i k_i$ : 有一只妖精从  $u_i$  号地点沿道路飞到了  $v_i$  号地点, 沿途的所有地点 (包括起点与终点) 都会新长出  $k_i$  ( $1 \leq k_i \leq 10^7$ ) 朵颜色为  $c_i$  ( $1 \leq c_i \leq 10^9$ ) 的花。保证  $u_i$  与  $v_i$  号地点存在。
- 3  $u_i c_i$ : 魔理沙想知道, 在  $u_i$  号地点的子树中有多少朵花的颜色编号小于或等于  $c_i$  ( $1 \leq c_i \leq 10^9$ )。子树的定义参见题目描述。

另外, 为了确保你每次都能立刻回答魔理沙的询问, 你需要维护一个变量  $last$ , 代表上一个询问的答案对  $2^{31}$  取模后的值 (如果还没有进行过询问, 规定其值为 0)。每当你读入一个操作时, 除了第一个数之外的所有数都需要对  $last$  进行按位异或, 得到的结果才是对应的数的真实值。

两个非负整数  $A$  和  $B$  的按位异或结果 (记为  $A \oplus B$ ) 定义如下:

- 在  $A \oplus B$  的二进制表示中,  $2^k$  ( $k \geq 0$ ) 所对应的数位当且仅当  $A$  和  $B$  的二进制表示中对应位恰有一个 1 时等于 1, 否则等于 0。

输入保证:

- 输入中的每个数都是小于  $2^{31}$  的非负整数。
- 用上述方法得到的每个数的真实值都满足上面提到的各个限制。
- 地点总数不会超过  $5 \times 10^4$ 。
- 第二种操作的总数不会超过  $5 \times 10^4$ 。

## 输出

对每个第三种操作，在单独的一行内输出一个非负整数，代表  $u_i$  号地点的子树中颜色不大于  $c_i$  的花朵的数量。

## 样例

标准输入	标准输出
3 5	12
1 2	14
1 3	
2 2 3 1 4	
3 1 1	
1 13	
2 13 8 14 13	
3 13 14	

## 注释

样例输入的真实值应当为：

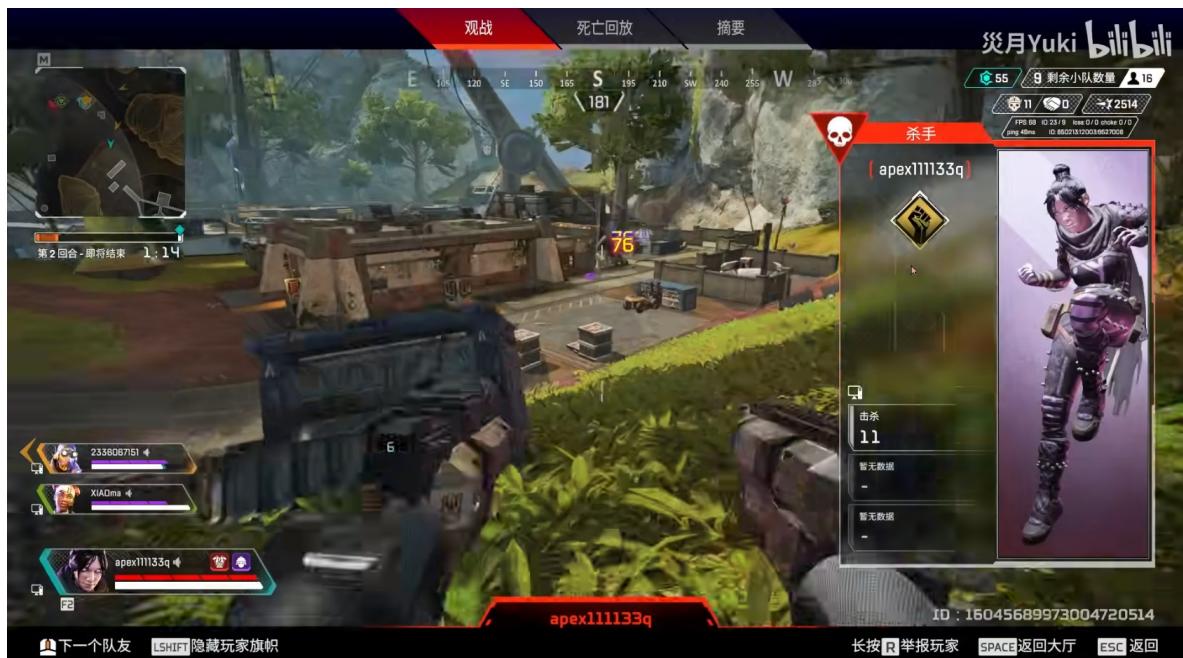
```
3 5
1 2
1 3
2 2 3 1 4
3 1 1
1 1
2 1 4 2 1
3 1 2
```

# 题目 G. Gua!

输入文件: 标准输入  
 输出文件: 标准输出  
 时间限制: 1 秒  
 内存限制: 1024 MB

Apex 英雄是一款知名的大逃杀游戏，游戏内容是：你逃，大哥杀（大哥：指使用作弊软件，即“外挂”的玩家）。

譬如，有一些大哥喜欢通过修改枪的参数来获得优势。如下图，你可以在风暴点里看到同时持有两把武器开火并且精准命中你的大哥，如天神下凡般给其他玩家带来巨大的心理阴影。



Bilibili: BV1V3411j7QW «风暴点里有什么，有圣枪游侠»

在输了一晚上排位以后，你被外挂的嚣张和游戏开发公司重生的懒惰给彻底激怒了。你打算写一封投诉信来教会重生如何鉴别外挂。

有一名玩家使用了一把每发子弹至多  $B$  点伤害的武器击杀了你。这把武器的射速是  $R$  RPM (每分钟射击数, Rounds Per Minute)，意思是该武器在射击一发子弹后必须等待至少  $1/R$  分钟才能进行下一次射击。当  $R = 0$  时，意味着这名玩家其实没有武器或者子弹用光了，因此不能造成伤害。

死亡回放显示，该玩家在  $S$  秒内对你造成了  $D$  点伤害，时间从第一发子弹命中算起到最后一发子弹命中为止。如果我们说他一定修改武器参数了，那么意思是他的伤害比至多可能的伤害还多。在这里，我们评判作弊时仅考虑提到的  $B, R, S$  三个因素，忽略掉游戏中的其他要素，比如弹匣容量、换弹时间、还有游戏特色 BUG。

我们使用图中的小帮手这把枪来举例。这把枪命中身体时能对防御型传奇造成 38 点伤害，而大哥在仅 1 秒内打出了  $152 = 38 \times 4$  点伤害。由于这把枪的射速是 156 RPM，在一秒内只可能等待  $156/60 = 2.6$  个射击间隔，也即最多射击 3 发子弹。因为  $4 > 3$ ，所以我们可以肯定他开了。

写一个给重生看的演示程序，来教会他们用  $B, R, D, S$  鉴挂。

## 输入

第一行包含一个整数  $T(1 \leq T \leq 10^3)$ ，表示数据组数。

对于每组数据，输入一行四个整数  $B, R, D, S(0 \leq B, R, D, S \leq 2000)$ ，表示最大单发子弹伤害，以 RPM 为单位的射速，玩家造成的伤害，以秒为单位的从第一颗子弹命中到最后一颗子弹命中之间的时间。

## 输出

对于每组数据输出一行,

- 如果一定修改武器参数了, 输出“gua!” (不含引号) ,
- 否则输出“ok” (不含引号) 。

## 样例

标准输入	标准输出
7	gua!
38 156 152 1	ok
280 25 280 0	ok
99 51 9 10	gua!
0 0 1 1	gua!
99 0 1 1	ok
11 1080 209 1	gua!
11 1080 210 1	

## 注释

第一组数据在前文中已有解释。

对于第二组数据, 克雷贝尔一枪爆头确实能造成这么多伤害, 而此时我们将一发子弹的时间计为 0。也许大哥开了自动瞄准, 但我们通过给定的信息无法做出判断。

第三组数据中, 这位青铜玩家展现出了非凡的枪法。需要注意的是, 这里子弹并没有造成最大伤害, 这可能是因为使用和平捍卫者等霰弹枪命中了部分弹丸。只要总伤害量没有超过最大可能的伤害, 我们就无法判定是否真的开挂。

第四和第五组数据中, 玩家其实没有武器或者子弹用光了, 常理来讲不能造成伤害——除非使用高科  
技。

## 题目 H. Heirloom Painting

输入文件: 标准输入  
输出文件: 标准输出  
时间限制: 2 秒  
内存限制: 1024 MB

艺术大师咖波有一个绘画机器人，用来帮助他制作一些精美的艺术品。

这天，咖波画了一个圆环，并将其分割成了  $n$  个格子。咖波总共调制了  $m$  种颜色，他希望用这些颜色给圆环涂上色，从而画成一件新的艺术品。然而，当初出于成本的考虑，他的绘画机器人有很多技术上的限制，比如的喷嘴是一个年久失修的传家宝，这导致他的机器人每次只能给连续的恰好  $k$  个格子涂上颜色。幸好机器人使用的颜料并不会因为重复覆盖同一个格子而混合——后面涂上的颜色将取代此前的颜色。

现在，咖波想知道他的机器人最少需要涂多少次颜色才能将这个圆环画成他想要的样子。当然，也有可能他的笨笨机器人其实没法完成任务。

### 输入

第一行输入一个整数  $T$  ( $1 \leq T \leq 10^5$ )，表示测试数据组数。

对于每组数据，第一行包含三个整数  $n, m$  和  $k$  ( $1 \leq n, m \leq 10^6, 1 \leq k \leq n$ )，分别表示圆环的格子数，颜色的种类数和机器人单次涂色的长度。第二行包含  $n$  个用空格分开的整数  $c_1, c_2, \dots, c_n$  ( $1 \leq c_i \leq m$ )，其中  $c_i$  表示在咖波想要将第  $i$  个格子涂成的颜色。

需要注意的是，画布上一开始是没有颜色的。你可以认为没有颜色的格子的颜色是  $-1$ 。

数据保证  $n$  的总和不超过  $10^6$ 。

### 输出

对于每组数据，输出一行包含一个整数，表示机器人最少涂色次数。如果机器人无法完成任务，输出  $-1$ 。

### 样例

标准输入	标准输出
3 11 4 2 1 1 1 2 2 3 3 3 4 4 1 5 2 1 1 2 1 2 1 6 2 2 1 2 1 2 1 2	6 5 -1

### 注释

对于第一组数据，一个最优涂色方案是：

1. 将第 11 个和第 1 个格子涂成颜色 1。注意，由于是圆环，所以第 11 个格子和第 1 个格子是相邻的。
2. 将第 2 个格子和第 3 个格子涂成颜色 1。
3. 将第 4 个格子和第 5 个格子涂成颜色 2。
4. 将第 6 个格子和第 7 个格子涂成颜色 3。
5. 将第 8 个格子和第 9 个格子涂成颜色 3。
6. 将第 9 个格子和第 10 个格子涂成颜色 4。注意，此时第 9 个格子的颜色将从颜色 3 变成颜色 4。

## 题目 I. It Takes Two of Two

输入文件: 标准输入  
输出文件: 标准输出  
时间限制: 1 秒  
内存限制: 1024 MB

独学而无友, 则孤陋而寡闻。



一把可以坐的椅子和一只早柚。

作为世界上知名的椅子生产商 Mihomo 的领导, 田老师鼓励公司员工之间进行合作, 旨在扩大生产线的规模。为了保证合作的质量, 所有合作必须遵守三条规则:

- 每个合作关系涉及 2 名不同的员工。
- 不存在两个合作关系涉及同一对员工。
- 每名员工只能在不超过 2 个不同的合作关系里。

公司有  $n$  名员工, 包括田老师本人。田老师认为公平非常重要, 所以他决定掷一些代表公平的骰子来决定分配。“掷骰子”的程序如下:

- 起初, 合作关系集合  $R$  初始化为空 ( $\emptyset$ ) , 随后将进入循环。
- 在一次循环中, 从  $1, 2 \dots, n$  中独立均匀随机地选择两个整数  $u, v$ 。如果  $R \cup (u, v)$  满足上述规则, 将  $(u, v)$  添加到  $R$  里, 否则保持  $R$  不变。
- 在一次循环前, 如果不存在可能被加进  $R$  的合作关系, 程序会立刻结束。

田老师花了 114 秒写了一个小程序来完成上述过程, 但他发现程序运行了整整 514 秒。由于田老师对于九以上的算数都不太熟练, 他希望你能帮他计算一下期望下这个程序需要循环多少次, 来判断是不是程序写错了。

### 输入

输入一行一个整数  $n$  ( $1 \leq n \leq 200$ ), 表示公司的员工数量。

### 输出

输出一个十进制实数表示你的答案。

如果你的答案与参考答案之间的相对或绝对误差小于或等于  $10^{-6}$ , 你的答案将被判定为正确。

为了输出小数点后若干位, 以 9 位数字为例, 你可以使用:

- C 格式输入输出: `printf("%.9lf\n", ans)` 或 `printf("%.9Lf\n", ans)`
- C++: `cout << fixed << setprecision(9) << ans << endl`
- Python: `print("%.9f" % ans)`

## 样例

标准输入	标准输出
1	0.000000000
2	2.000000000
3	8.250000000

## 注释

对于第一组样例，田老师无法和自己合作，因此程序会立即终止。

对于第二组样例，最终  $R$  集合只能是  $\{(1, 2)\}$  和  $\{(2, 1)\}$  中的一种，同时每次会有  $1/2$  的概率随机到不可行的  $(1, 1), (2, 2)$ 。因此，期望下的循环次数是

$$1 \times (1/2) + 2 \times (1/4) + 3 \times (1/8) + \dots = 2.$$

## 题目 J. Just Some Bad Memory

输入文件: 标准输入  
输出文件: 标准输出  
时间限制: 1 秒  
内存限制: 1024 MB

深呼吸, 让我以最快的速度告诉你需要做什么。

给定一个简单无向图  $G = (V, E)$ , 你需要告诉我最少需要加几条边, 才能得到一个既有奇环也有偶环的简单无向图。

简单图是没有重边和自环的图, 意味着每条边连接两个不同的顶点, 并且没有两条边连接同一对顶点。

一个环定义为一列两两不同的顶点  $\{v_1, v_2, \dots, v_k\}$ , 满足  $(v_i, v_{i \bmod k+1}) \in E$ 。 奇和偶指的是  $k$  的奇偶性。最小的奇环大小为 3, 最小的偶环大小为 4。

### 输入

输入第一行两个正整数  $n, m$  ( $1 \leq n \leq 10^5, 0 \leq m \leq \min\{2 \times 10^5, \binom{n}{2}\}$ ), 表示顶点数 ( $|V|$ ) 和边数 ( $|E|$ )。

接下来  $m$  行, 每行两个整数  $u, v$  ( $1 \leq u, v \leq n, u \neq v$ ), 表示有一条边连接  $u$  和  $v$ .

输入保证形成一个简单图。

### 输出

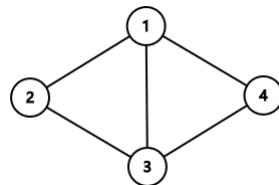
输出一行表示你的答案。如果无法完成目标, 输出 ‘-1’。

## 样例

标准输入	标准输出
3 3 1 2 2 3 1 3	-1
4 0	5
5 4 1 2 2 3 3 4 4 5	2
4 6 1 2 1 3 1 4 2 3 2 4 3 4	0
4 4 1 2 2 3 3 4 4 1	1
7 7 1 2 2 3 3 4 4 1 5 6 6 7 7 5	0

## 注释

以下图示表示了样例 2 的一种最优解。其中，奇环有  $\{1, 2, 3\}$  和  $\{1, 3, 4\}$ ，唯一的偶环是  $\{1, 2, 3, 4\}$ .



## 题目 K. Known as the Fruit Brother

输入文件: 标准输入  
输出文件: 标准输出  
时间限制: 4 秒  
内存限制: 1024 MB

在游戏《英雄联盟》中，爆裂球果是一种可以爆炸的植物，在爆炸时可以把范围内的角色弹飞若干米远。



Cryin 选手曾使用爆裂球果来加速他和他队友的移动。

现在假设你的角色在召唤师峡谷中。为了简单起见，

- 召唤师峡谷可以被视为一个无限的二维平面，而你的角色是一个忽略半径的点。
- 所有的障碍物都是与坐标轴平行的矩形。你不能进入障碍物内部，但是可以沿着边界行动。
- 爆裂球果的使用方法是：你的角色必须走到爆裂球果所在的坐标，破坏它，并且你可以跳到距离不超过  $R$  的范围内的任意位置（除非在障碍物内部）。你可以假设这个过程是瞬时的，也即不会花费任何时间。

现在地图上有  $n$  个矩形障碍物和  $m$  个爆裂球果，而你希望从  $(x_s, y_s)$  出发到  $(x_t, y_t)$ 。你的角色的移动速度是每秒 1 个单位。你能计算出从起点到终点的最短时间开销吗？

### 输入

第一行包含三个整数  $n, m, R$  ( $0 \leq n, m \leq 40, 1 \leq R \leq 10^6$ )，分别表示矩形障碍物的数量、爆裂果实的数量和跳跃半径。

随后  $n$  行，第  $i$  行包含四个整数  $x_{i,1}^b, y_{i,1}^b, x_{i,2}^b, y_{i,2}^b$  ( $-10^6 \leq x_{i,1}^b, y_{i,1}^b, x_{i,2}^b, y_{i,2}^b \leq 10^6, x_{i,1}^b < x_{i,2}^b, y_{i,1}^b < y_{i,2}^b$ )，表示第  $i$  个障碍物的左下坐标是  $(x_{i,1}^b, y_{i,1}^b)$ ，右上坐标是  $(x_{i,2}^b, y_{i,2}^b)$ 。

随后  $m$  行，第  $i$  行包含两个整数  $x_i^c, y_i^c$  ( $-10^6 \leq x_i^c, y_i^c \leq 10^6$ )，表示第  $i$  个爆裂球果的位置是  $(x_i^c, y_i^c)$ 。

最后一行包含四个整数  $x_s, y_s, x_t, y_t$  ( $-10^6 \leq x_s, y_s, x_t, y_t \leq 10^6$ )，表示起点  $s$  和终点  $t$ 。

输入保证：

- 任何两个障碍物（包括边角）严格不交。
- 爆裂果实的位置、起点  $s$  和终点  $t$  两两不同，并且均严格处于障碍物外。

### 输出

输出一行一个十进制实数，表示从起点到终点的最短时间开销。如果你的答案与参考答案之间的相对或绝对误差小于或等于  $10^{-6}$ ，你的答案将被判定为正确。

可以证明, 一定能从起点到达终点。

为了输出小数点后若干位, 以 9 位数字为例, 你可以使用:

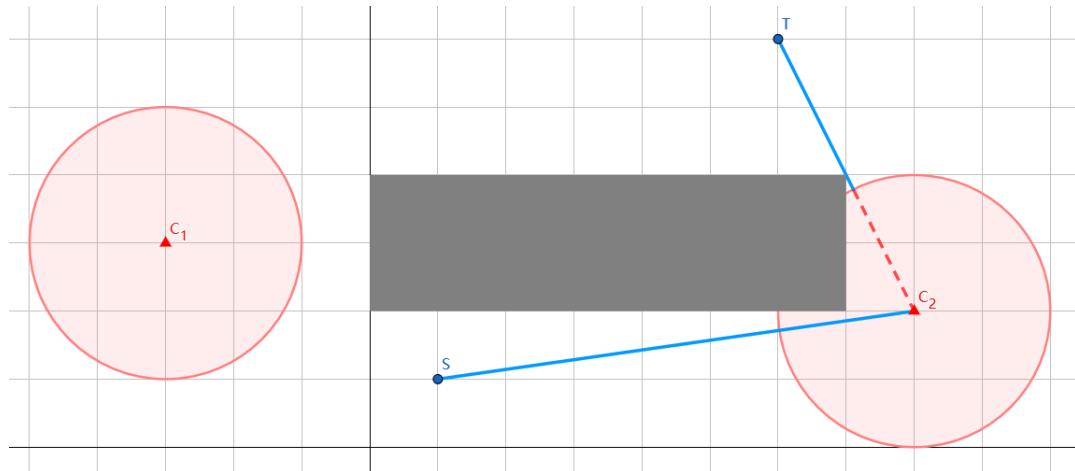
- C 格式输入输出: `printf("%.9lf\n", ans)` 或 `printf("%.9Lf\n", ans)`
- C++: `cout << fixed << setprecision(9) << ans << endl`
- Python: `print("%.9f" % ans)`

## 样例

标准输入	标准输出
1 2 2 0 2 7 4 -3 3 8 2 1 1 6 6	9.543203767

## 注释

样例输入的图示如下:



答案是  $\sqrt{7^2 + 1^2} + \sqrt{2^2 + 4^2} - 2 \approx 9.543203767$ 。

## 题目 L. Last Warning of the Competition Finance Officer

输入文件: 标准输入  
输出文件: 标准输出  
时间限制: 5 秒  
内存限制: 1024 MB

人年纪大了就会安于一隅，例如孔乙己会固守他迂腐的思想，这是自然的——更何况从大秦帝国来的少府<sup>1</sup>大人。少府大人向往自然，不喜欢充满了手机信号和自来水的现代生活，因此他选择了天然核子温泉度假村来举办传统的复活鸡总决赛<sup>2</sup>。与此同时，他的说话的方式也很怪异。比如，少府大人认为不合他本人心意的事物是不合礼教的，此时他也许会这样抗议：“侵犯名誉权！！！侵犯名誉权！！！侵犯名誉权！！！”

人们将复活鸡总决赛的参赛选手称为猎鸡人，而鸡是需要一些特定的知识来猎杀的。例如，许多选手正在研究如何使用概率和构造方法来击杀小 boss “哈希鸡”；而头号种子芙丽题安已经在计划击杀终极大 boss “核子可乐鸡”了。为此，他希望用机器学习（噢我的上帝，少府大人最讨厌这些东西了）来分析少府大人的语录，这样他就可以通过模仿少府的腔调来获得管理权限，潜入核子可乐鸡的藏匿地点并最终击杀大 boss 得到冠军。

少府所说过的话可以被看作一个仅包含小写字母的字符串  $s$ 。芙丽题安已经筛选出  $n$  个短语  $\{t_i\}$  (同样由小写字母构成)作为少府最经典的语录。对于每个  $t_i$ ，有一个对应的权值  $v_i$ ，表示这个短语有多典。

为了模仿少府，芙丽题安希望计算出  $s$  的每一个前缀的典值。一个字符串的典值由如下方式定义：

- 字符串  $u$  的典值，等于其所有不同的成分的权值之和。
- 字符串  $u$  的一个成分，是  $u$  的一个不相交子串的集合，且集合中每一个子串都在语录中出现。
- 一个成分的权值，等于集合中所有子串对应权值的乘积。空集的权值定义为 1。
- 两个  $u$  的子串是不同的，当且仅当它们在  $u$  中出现的位置不同。因此，语录中的一个单词可以在字符串的一个成分中出现多次，并且它们都将被计入权值乘积的贡献。
- 两个  $u$  的成分是不同的，当且仅当某个子串在仅在一个成分中存在。

由于答案可能会特别大，请输出答案对 998 244 353 取模后的结果。

### 输入

第一行输入一个仅包含小写字母的字符串  $s$  ( $1 \leq |s| \leq 2 \times 10^5$ )，含义见题目描述。

第二行输入一个正整数  $n$ ，表示语录中单词的个数。

此后  $n$  行，第  $i$  行包含一个仅包含小写字母的字符串  $t_i$  和一个整数  $v_i$  ( $1 \leq |t_i| \leq 2 \times 10^5$ ,  $0 < v_i < 998 244 353$ )，用空格隔开，分别表示语录的第  $i$  个单词和其权值。

输入保证  $t_i$  两两不同，且它们长度总和小于等于  $2 \times 10^5$ 。

### 输出

输出一行用空格隔开的  $|s|$  个整数，第  $i$  个表示  $s$  的长度为  $i$  的前缀的典值对 998 244 353 取模的结果。

<sup>1</sup>少府，秦官名，九卿之一，专门负责管理皇家的财政收入。战国时韩始设，秦汉沿置。

<sup>2</sup>Easter Chicken Finals

## 样例

标准输入	标准输出
ababa 2 aba 2 ba 3	1 1 6 6 26
qfmyqqfmyqqfmyq 2 qfmyq 111111 myqq 404968002	1 1 1 1 111112 405079114 405079114 405079114 405079114 771912310 239058268 239058268 239058268 239058268 31169271
wwwsoupu.net.com 2 money 999999 soup 998244352	1 1 1 1 1 1 0 0 0 0 0 0 0 0

## 注释

对于第一个样例的整个字符串，有如下成分：

- ababa 6 = 2 × 3
- aba.. 2
- ..aba 2
- .ba.. 3
- ...ba 3
- .bab 9 = 3 × 3
- ..... 1

其中，未使用的字符用点来代替，不同的子串用下划线和上划线标出。

因此，全串的典值为

$$(6 + 2 + 2 + 3 + 3 + 9 + 1) \bmod 998\,244\,353 = 26.$$

请注意第二个样例的输出因 PDF 版面原因做了格式调整，实际上并无额外换行。

## 题目 M. My University Is Better Than Yours

输入文件: 标准输入  
 输出文件: 标准输出  
 时间限制: 2 秒  
 内存限制: 1024 MB

大家总喜欢搞些什么排行榜。有一说一, 排行榜通常不重要, 除非——比如你老板要你做一下年终总结。

为了实现建设世界一流大学和建设世界一流学科的目标, 不少大学都用各种方式提升排名: 发表论文、申请基金、提升多样性...不过看起来这些不容易, 而且 US News 和 Times 这样的机构并不一定会公正评判你的工作。因此, 一些大学更聪明——自己发布排行榜, 这可以使得自己的名次间接变好。比如, 通过上海某大学发布的软科排名 (ARWU) 作为桥梁, 咖波甚至可以论证他的闵行理工学院要好于麻省理工学院:

### My University Is Better Than Your University

My University
Your University

---

zizhengfang.com/applets/transitivity

<b>QS</b>	Shanghai Jiao Tong University (#46)	>	Washington University in St. Louis (#118)
<b>ARWU</b>	Washington University in St. Louis (#23)	>	Zhejiang University (#52)
<b>QS</b>	Zhejiang University (#42)	>	University of California - Los Angeles (#44)
<b>ARWU</b>	University of California - Los Angeles (#14)	>	University of Pennsylvania (#15)
<b>QS</b>	University of Pennsylvania (#13)	>	University of California - Berkeley (#27)
<b>ARWU</b>	University of California - Berkeley (#5)	>	California Institute of Technology - Caltech (#9)
<b>Times</b>	California Institute of Technology - Caltech (#2)	>	Massachusetts Institute of Technology - MIT (#5)

Therefore, Shanghai Jiao Tong University is better than Massachusetts Institute of Technology - MIT.

[Flip](#)

[Share](#)

<https://www.zizhengfang.com/applets/transitivity>

当然, 除了咖波的工作面试里, 上面这种比较最多出现在知名公众号二月十三里。与此同时, 咖波想知道在这种比较方式下排名会发生什么样的变化。考虑  $m$  个排行榜中的  $n$  所大学, 为简单起见, 所有大学都用一个从 1 到  $n$  的数字编号来表示。这里, 咖波定义学校  $x$  直接好于学校  $y$ , 当且仅当存在一份大学排名中学校  $x$  排在学校  $y$  前面。如果咖波认为学校  $x$  好于学校  $y$ , 当且仅当存在一个学校的序列  $\{s_1, s_2, \dots, s_k\}$  ( $k \geq 2$ ), 满足:

- $s_1 = x, s_k = y$ ;
- $\forall i \in \{1, 2, \dots, k-1\}$ , 学校  $s_i$  直接好于学校  $s_{i+1}$ 。

对于每一所学校, 咖波想知道这所学校好于多少所其他学校。

## 输入

第一行包含两个整数  $n$  和  $m$  ( $1 \leq n \leq 5 \times 10^5$ ,  $1 \leq m \leq 5 \times 10^5$ ,  $1 \leq n \times m \leq 10^6$ ), 表示大学的数量和大学排名的数量。

随后  $m$  行, 每行  $n$  个互不相同的整数, 用空格隔开。第  $i$  行的内容  $s_{i,1}, s_{i,2}, \dots, s_{i,n}$  ( $1 \leq s_{i,j} \leq n$ ) 表示第  $i$  个学校排名, 顺序从高到低。

## 输出

输出一行用空格隔开的  $n$  个整数  $a_1, a_2, \dots, a_n$ , 第  $i$  个数  $a_i$  表示学校  $i$  好于多少所其他学校。

## 样例

标准输入	标准输出
4 2 1 2 3 4 1 3 2 4	3 2 2 0
4 2 1 2 3 4 4 3 2 1	3 3 3 3
5 2 5 4 3 2 1 5 4 3 2 1	0 1 2 3 4

## 题目 N. Nine Is Greater Than Ten

输入文件: 标准输入  
输出文件: 标准输出  
时间限制: 1 秒  
内存限制: 1024 MB

知名数学家、哲学家田老师正在有力论证

$$9 > 10$$

是正确的。

证明. 我们写下两个数的十进制表示:

9.

10

我们观察这两个数字从前到后第一位不同的数字, 由于  $9 > 1$ , 因此  $9 > 10$ . *Q.E.D.*

同样, 你可以很容易地用同样的论证来证明  $114\ 514 < 1919$ :

114514

1919..

和  $9 < 999$ :

9..

999

哇, 数学! 真有逻辑! 现在你已经完全理解了田老师是如何对两个数字进行比较的。请写一个程序, 来比较两个输入的数字吧!

### 输入

输入一行用空格隔开的两个整数  $a, b$  ( $1 \leq a, b \leq 20\ 220\ 924$ ), 均没有前导零。

### 输出

使用田老师的方式比较,

- 如果  $a > b$ , 输出 “ $a>b$ ” (不含引号) 。
- 如果  $a < b$ , 输出 “ $a<b$ ” (不含引号) 。
- 如果  $a = b$ , 输出 “ $a=b$ ” (不含引号) 。

### 样例

标准输入	标准输出
9 10	9>10
114514 1919	114514<1919
9 999	9<999
99 99	99=99
2022 924	2022<924