

第三届环球杯



第 0 阶段：美国（试验竞赛）

2024 年 6 月 1-2 日

该问题集应包含 13 个问题，共 21 页编号。

根据



国际大学生程序设计竞赛（ICPC）

主办方





问题 A.发展停滞

日

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 2048 兆字节

你现在负责两名编程实习生，必须开发一个大型系统。有许多任务需要在暑期结束前完成。您知道每个实习生完成每项任务需要多长时间（以分钟为单位）。
计算完成系统开发的所有任务所需的最少分钟数，假设只有两名实习生负责开发，他们同时独立工作，不分担任务，实习生完成所有任务所需的时间是逐个完成每个任务所需的分钟数的总和。

输入

第一行输入包含一个整数 n ($1 \leq n \leq 50$)，即任务数。
接下来的 n 行中，每一行都包含两个整数 a 和 b ($1 \leq a, b \leq 10^5$)。每行代表一个任务，其中 a 是第一个实习生完成任务所需的分钟数， b 是第二个实习生完成任务所需的分钟数。

输出

输出一个整数，即完成开发项目所需的最少分钟数。

实例

标准输入	标准输出
4 100 1 1 90 1 20 1 20	3
2 314 1 592 6	7



问题 B.Champernowne 子串

输入文件：标准输入
输出文件：标准输出
时间限制：10 秒 内存限制：2048 兆字节

Champernowne 字符串是由正整数的基 10 表示依次串联而成的无限字符串。
开头是 1234567891011121314...
可以证明，任何有限的数字字符串都至少会作为子串出现在钱姆诺恩字符串中一次。
给定一个由数字和问号组成的字符串，通过用 0 到 9 之间的一个数字替换每个问号，计算该字符串作为子字符串出现在 Champernowne 字符串中的最小可能索引。每个问号可以对应不同的数字。
由于这个索引可能很大，因此打印出它的模数 998 244 353。

输入

第一行输入包含一个整数 t ($1 \leq t \leq 10$)，即测试用例数。
接下来的 t 行中，每一行都包含一个字符串 s ($1 \leq |s| \leq 25$)，由 0 到 9 的数字或问号组成。

输出

输出 t 行。按顺序为每个测试用例输出一行，其中包含一个整数，即该字符串可能作为子串出现在 Champernowne 字符串中的最小索引，模数为 998 244 353。

示例

标准输入	标准输出
9	11
0	7
???1	14
121	10
1?1?1	314159
??5?54?50?5?505?65?5	796889014
000000000000	7777
?2222222	8058869
?3???????9?98???????1??0	38886
9?9??0???????????2	



问题 c. 比较器

日

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

2048 兆字节

许多编程语言都允许您定义自定义比较器来对用户定义的对象进行排序。

IFY 是一种编程语言，其中唯一的程序是用作比较器的函数。这些函数对位字符串（以下称为 "字"）进行操作。字从最左边的字符开始按 1 索引。

要在 IFY 中编写一个关于两个字的函数，程序员需要编写一系列 if 语句。每个 if 语句从第一个字中选择一个位，从第二个字中选择一个位，然后对这两个位上的某个表达式进行求值。如果表达式在选择的位上求值为 1，则函数立即返回指定的返回值。否则，将执行到下一个 if 语句，不返回任何内容。

每个 if 语句的语法是 $a\ b\ expr\ r$ ，其中整数 a 指定第一个字中的一个位，整数 b 指定第二个字中的一个位， $expr$ 是使用变量 x 和/或 y 的布尔表达式， r 是表达式求值为 1 时从函数返回的值（0 或 1）。例如，考虑 if 语句 "2 3 $x|y$ 0"。该表达式的意思是：如果第一个字的第二位或第二个字的第三位被设置，则返回 0，否则继续下一条语句。

有效表达式的语法如下

- x 、 y 、0 和 1 都是有效的表达式。
- 如果 E 是有效表达式，那么 (E) 也是有效表达式。
- 如果 E 是有效表达式，那么 $!E$ 也是有效表达式。
- 如果 $E1$ 和 $E2$ 是有效表达式，那么所有形式为 $E1\oplus E2$ 的字符串都是有效表达式，其中 \oplus 是 =（等于）、&（和）、|（或）或 ^（xor）中的一种。

括号的优先级最高，其次是一元运算符 $!$ ，然后依次是二元运算符 =、&、| 和 ^。所有二元运算符都是左关联运算符。表达式中没有空格。

下面是每个运算符的真值表：

x	$!x$
0	1
1	0

Рис. 1: 唯一一元运算符的真值表。



日								
x	y	$x=y$	x	y	$x\&y$	x	y	$x\hat{y}$
0	0	1	0	0	0	0	0	0
0	1	0	0	1	0	0	1	1
1	0	0	1	0	0	1	0	1
1	1	1	1	1	1	1	1	0

Рис.2: 二进制算子的真值表。

要使函数 $f(x, y)$ 作为比较器正常运行，它必须满足某些属性。例如，样本 1 是对长度为 2 的位字符串进行排序的有效比较器。



从形式上看，比较器必须满足的三个属性是反身属性、对称属性和传递属性，具体如下：

- 1. **反身性**：对于所有 x ， $f(x, x)$ 都应返回 0。
- 2. **对称**：对于所有 x 、 y ，如果 $f(x, y)$ 返回 1，那么 $f(y, x)$ 必须返回 0。
- 3. **传递性**：对于所有 x 、 y 、 z ，如果 $f(x, y)$ 和 $f(y, z)$ 都返回 1，那么 $f(x, z)$ 也必须返回 1。

给定一个 IFY 语言中的函数，通过计算违反这三个属性的频率来确定该函数满足这三个属性的程度。首先，在给定长度的所有单词中，计算反身属性失效的单词数。接着，统计对称属性失效的词对数量。最后，计算反式属性失效的三对单词的数量。

输入

第一行输入包含两个整数 n ($0 \leq n \leq 2 \cdot 10^5$) 和 k ($1 \leq k \leq 10$)，其中 n 是函数的行数， k 是被比较值的位数。

接下来的 n 行中，每行都包含四个 $a\ b\ expr\ r$ 形式的标记，其中 a 和 b ($1 \leq a, b \leq k$) 是 x 和 y 中要考虑的位的索引， $expr$ 是一个有效的表达式， r ($0 \leq r \leq 1$) 是返回值。如果没有触发 if 语句，最后一行给出返回值。所有表达式的总长度最多为 10^6 。

输出

单行输出三个空格分隔的整数。第一个整数是违反了反身性的单词数，第二个整数是违反了对称性的成对单词数，第三个整数是违反了反身性的成对单词数。

实例

标准输入	标准输出
3 2 1 1 (x=0)&(y=1) 1 1 1 (x=1)&(y=(x^x))0 2 2 (x=1) (y=0) 0 1	0 0 0
4 3 2 1 x=0&(y=1) 1 1 2 !x^!!! y 0 2 3 ((x 1)=y)&1&1 1 3 1 !x&!x&!x 0 1	3 25 52



问题 D. 多面群

日

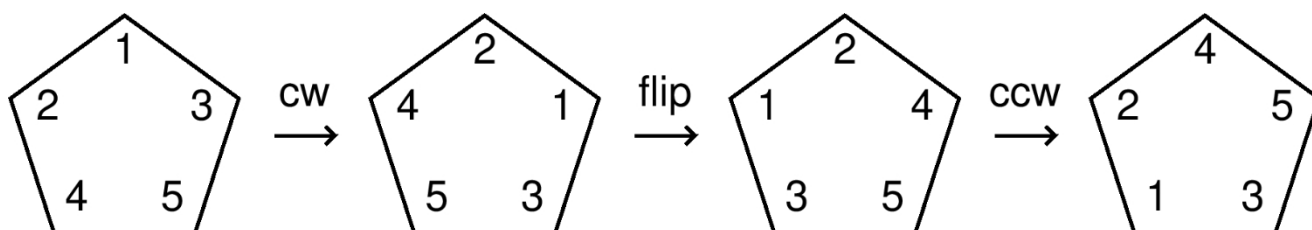
输入文件：标准输入

输出文件：标准输出

时间限制 1 秒 内存限制

2048 兆字节

在数学中，二面群 D_n 是正 n 形的对称群。旋转和反射是 D_n 的元素，事实上，二面群的所有元素都可以表示为一系列旋转和反射。 D_n 的元素通过置换 n 形的顶点来作用于 n 形。例如，考虑一个正五边形，其顶点的初始标记为 1、3、5、4、2（顺时针方向，从顶点开始）：



对五边形进行上述三个二面性操作（旋转、反射和再旋转）后，五边形顶点的重新标记如下：

$$1, 3, 5, 4, 2 \rightarrow 2, 1, 3, 5, 4 \rightarrow 2, 4, 5, 3, 1 \rightarrow 4, 5, 3, 1, 2.$$

给你一个用 1 到 n 的整数对正多边形的顶点进行任意顺时针标注的序列，以及第二个要测试的序列。请判断是否有可能对正多边形进行一系列二面性操作，从而使测试序列在变换后的多边形上显示为连续的顺时针顶点标注序列。

输入

第一行输入两个整数 n 和 m ， $(1 \leq m \leq n \leq 5 \cdot 10^4)$ 其中 n 是多边形的顶点数， m 是待测序列的长度。

下一行包含 n 个空格分隔的整数 d ($1 \leq d \leq n$)。这是多边形顶点的初始标记。保证从 1 到 n 的每个整数都正好出现一次。

下一行包含 m 个空格分隔的整数 t ($1 \leq t \leq n$)。这就是要测试的序列。

输出

输出一个整数，如果对初始多边形进行一系列二面性操作后，测试序列可以作为顶点标签的连续序列出现，则该整数为 1，否则为 0。



实例

日

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



问题 E. 房屋解构

日

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
2048 兆字节

在 "圆环乐园 "里，有一个圆，圆周上的点的间距相等。任何两个相邻点之间的距离都是 1。

圆圈的点上有人和房子。每个点上都有一个人、一间空房子，或者什么都没有。每个人都想去不同的房子。每个房子里最多只能有一个人。人们只能沿着圆周行走，不能穿过圆周。

目前，房子的数量多于人的数量，所以你想摧毁一些房子，使剩下的房子数量等于人的数量。假设你摧毁了一组房子 S ，让 $f(S)$ 成为每个人到达不同的未被摧毁的房子所需的最小步行总量。

计算 $f(S)$ 的最小值，计算有多少套房屋 S 达到了这个最小值。由于 S 的套数可能很多，因此输出它的模数 998 244 353。

输入

第一行输入包含三个整数 x 、 n 和 m ($1 \leq n < m \leq 2 \cdot 10^5$, $n + m \leq x \leq 10^9$)，其中 x 是圆周的点数， n 是人数， m 是房屋数。各点标记为 1 至 x ， x 点与 1 点相邻。

接下来的 $n + m$ 行各包含两个标记，一个整数 p ($1 \leq p \leq x$) 和一个字符 t ($t \in \{P, H\}$)，其中 p 表示人或房子的位置， t 表示点的类型，P 表示人，H 表示房子。所有位置都是不同的，位置将按递增顺序排列。

输出

输出两行。

第一行输出 $f(S)$ 的最小可能值。

第二行输出达到该最小值的集合 S 的数量，取模 998 244 353。

实例

标准输入	标准输出
6 2 4 1 P 2 H 3 P 4 H 5 H	2 3



第三届环球杯

第 0 阶段：美国（选拔赛），2024 年 6 月 1-2

6 H	日
1000000000 2 4	4
1 P	1
6 H	
31415926 H	
999999998 H	
999999999 H	
1000000000 P	



备注

日

对于第一个样本，最小总步行距离为 2，我们可以将房屋集破坏为 $\{2, 5\}$ 、 $\{4, 5\}$ 或 $\{5, 6\}$ 。

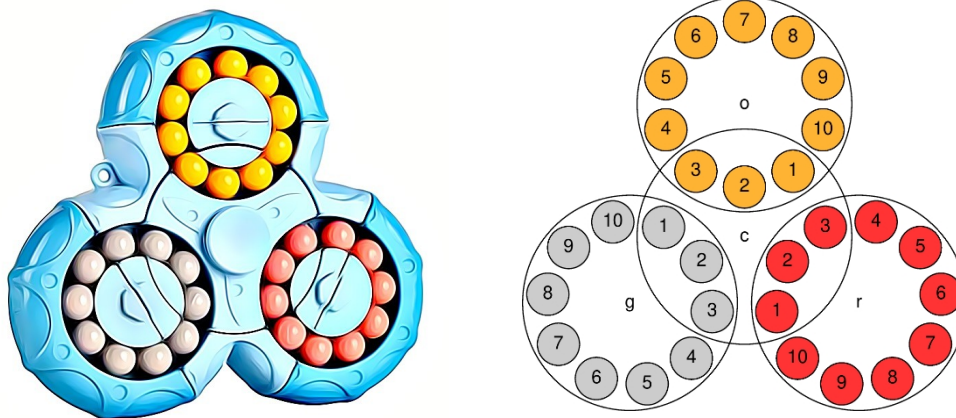
对于第二个样本，我们可以摧毁 $\{6, 31415926\}$ 处的一组房屋，最小总步行距离为 4。请注意，尽管最小步行距离可以通过多种方式实现，但由于摧毁的房屋集合相同，因此只计算一次。



问题 F. 魔豆

日

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：
2048 兆字节



魔豆 "是一个集小玩意和拼图于一体的玩具，有 30 颗彩色珠子在三个环形轨道上移动。在解锁状态下，上圆包含十颗无法区分的橙色珠子，左下圆包含十颗无法区分的灰色珠子，右下圆包含十颗无法区分的红色珠子。每个圆圈中的珠子都可以旋转。此外，中间的第四个圆圈也可以旋转，这样就可以在圆圈之间交换连续的三颗珠子。

你的弟弟刚刚借走了你的魔豆，并任意旋转了这些圆圈，把珠子弄乱了。你的任务是解开这个谜题，找到四个圆的有效旋转序列，从而回到已解状态。你不需要找到最短的解法，但是你的解法应该不超过 240 步。所提供的输入将是最多 240 次移动后的谜题状态。

输入

输入正好由三行组成。第一行描述上圆的十颗珠子，第二行描述左下圆，第三行描述右下圆。字符 o 代表橙色珠子，g 代表灰色珠子，r 代表红色珠子。珠子按顺时针顺序排列，并根据图示编号。

输入配置是在已解谜题上最多走 240 步的结果。

输出

输出一个整数 k ， $0 \leq k \leq 240$ ，即解题所需的步数。

接下来，如果 $k > 0$ ，则在 k 行上按顺序输出 k 次移动，每行一次。每次移动由一个字符组成，指定要旋转的圆，紧接着是一个数字，指定圆按顺时针方向旋转的距离。最上面的圆用 "o" 指定，左

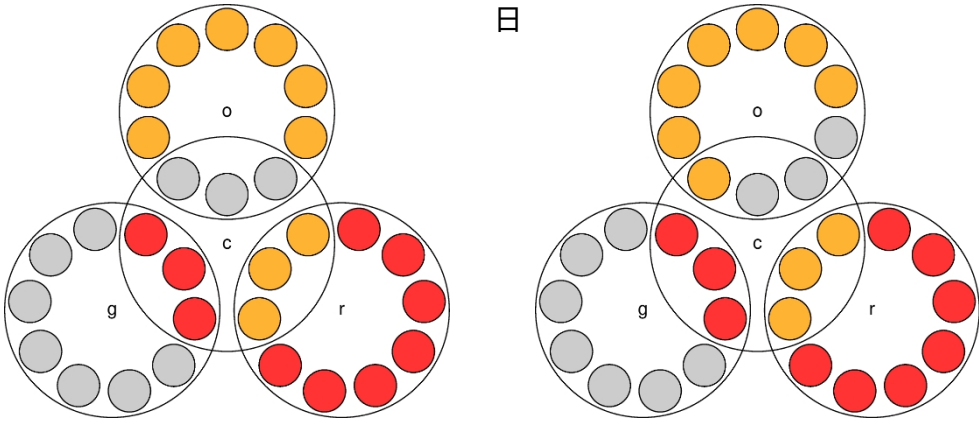


下方的圆用 "g "指定，右下方的圆用 "r "指定，中间的圆用 "c "指定。对于前三个圆，指定旋转的数字值为 1 到 9；对于中心圆，数字值为 1 到 2。

下面两张图片显示了两个样本的输入位置。



日



实例

标准输入	标准输出
ggg0000000 rrrggggggg g OOOrrrrrrrr	1 c2
眩晕 rrrggggggg g OOOrrrrrrrr	2 o1 c2



问题 G. 曼哈顿漫步

日

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
2048 兆字节

这是一个网格系统！你要从左上角开始，走到右下角。每个位置都是整数坐标，都有一个向下或向右的箭头和一个计时器，计时器每隔几秒钟就会将箭头从向下翻转到向右，或从向右翻转到向下。当你开始行走时，每个箭头都以相等的概率指向下方或右方，每个计时器都有一个实数值，这个实数值在从零到最大等待时间的范围内均匀选择。当计时器归零时，它会重置回最大值。

在任何时刻，只要你在某个地点，你就可以：

- 如果新位置在网格上，且箭头向下，则向下移动一个位置。
- 如果新位置位于网格上，且箭头指向右侧，则向右移动一个位置。
- 等待计时器倒计时结束，箭头就会从下往右或从右往下翻转。

当你到达一个新地点时，你可以看到计时器，因此在决定采取哪种行动时可以将其考虑在内。但是，您无法展望未来，只能看到您所在网格点的计时器和箭头。你讨厌等待，希望尽量减少等待箭头翻转的总时间。

如果以最佳方式做出决策，预计需要等待多长时间？

输入

单行输入包含三个整数 r 、 c ($1 \leq r, c \leq 10^3$) 和 p ($1 \leq p \leq 10^9$)，其中 r 是网格中的行数， c 是网格中的列数， p 是计时器能显示的最大值。

输出

输出一个数字，这是您做出最优决策时预计需要等待的时间。如果绝对误差或相对误差在法官答案的 10^{-6} 以内，您的答案将被接受。

实例

标准输入	标准输出
2 3 8	2.875
5 5 5	2.43223387

问题 H. MountainCraft

日

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

2048 兆字节

今天是电子游戏元世界 MountainCraft 中光辉灿烂的一天！正如它的名字一样，《MountainCraft》拥有一个广阔的开放世界，到处都是可供探索的山脉。你在游戏中的化身在一个偏僻的小岛上醒来，凝视着地平线上的群山。

地平线的视图可以用笛卡尔平面来模拟， x 轴 ($y = 0$) 将陆地和海洋分开。每座山峰都用一个点 (x, y) 表示，山峰两侧的斜率分别为+1 和-1，形成一个以山峰为顶点的三角形。

您的头像只能看到视口中的山脉部分。山脉的可见边缘（不与其他山脉重叠的部分）会以粗体显示。如果山脉重叠，重叠部分不会以粗体显示。山体重叠的边缘不一定相交。

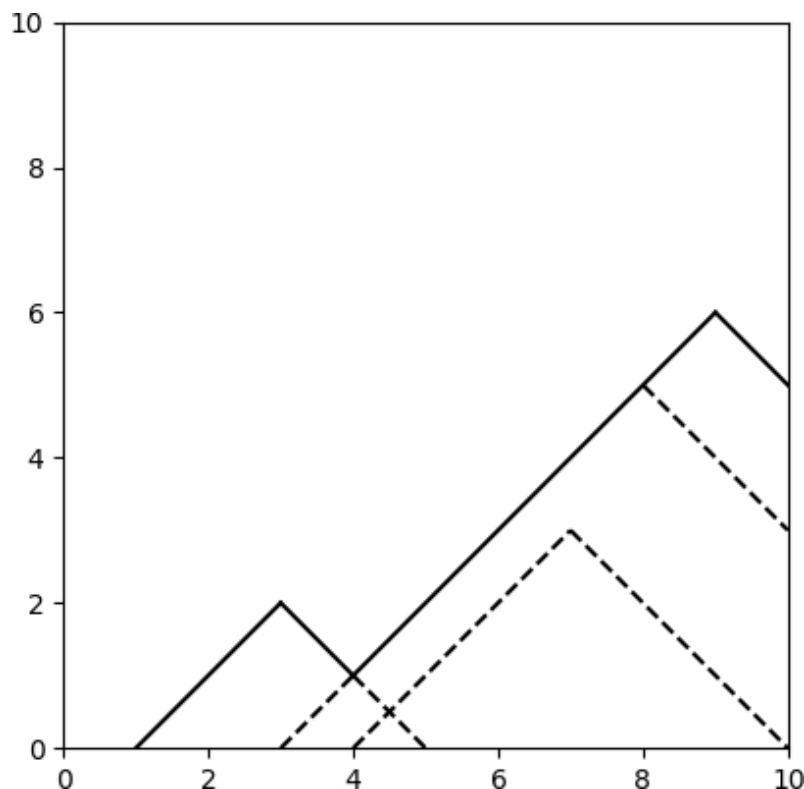


Рис. 3: 所有山脉出现后的第一个输入样本。

不幸的是，由于图形故障，山脉可能会出现和消失。每次更改后，您都想知道当前视口中可见的



所有粗体线条的总长度。

日

输入

第一行输入包含两个整数 q ($1 \leq q \leq 2 \cdot 10^5$) 和 w ($1 \leq w \leq 10^9$)，其中 q 是查询次数， w 是视口宽度。视口宽度从 0 到 w 不等。



接下来的 q 行中，每一行都包含两个整数 x ($0 \leq x \leq w$) 和 y ($0 < y \leq 10^9$)，即某座山的山峰。如果 (x, y) 是可见山峰的山顶，那么这座山将消失。否则，该山峰将变得可见。

输出

输出 q 行。对每个查询按顺序输出一行，并用一个数字表示，该数字是所呈现的粗体行的总长度。如果绝对误差或相对误差在法官答案的 10^{-6} 以内，您的答案将被接受。

实例

标准输入	标准输出
3 10	5.656854
3 2	12.727922
7 3	12.727922
9 6	
5 100	101.823376
31 41	120.208153
59 26	73.539105
31 41	0.000000
59 26	101.823376
31 41	

备注

在第一个示例中，前两座山相交于点 $(4.5, 0.5)$ 。

在第二个样本中，左边的山出现，然后消失，接着又出现。右边的山出现，然后消失。



问题 I. 不是另一种建设性

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：2048 兆字节

厌倦了解决几何问题，你决定解决下面的构造问题：找出一个长度为 n 的字符串，其中正好包含 NAC 的 k 个不一定连续的子序列。

不过，这个问题似乎太熟悉了。这里有一个转折点--你的朋友给了你部分字符串，所以你必须填上剩余的字符！

输入

第一行输入包含两个整数 n ($1 \leq n \leq 40$) 和 k ($0 \leq k \leq 2\,500$)，其中 n 是字符串的长度， k 是输出必须包含的不一定连续的 NAC 子序列的个数。

第二行包含一个长度正好为 n 的字符串，仅由大写字母和/或问号组成。

输出

输出一个大写字母字符串，用一个大写字母替换输入字符串中的每个问号，这样得到的字符串就正好有 k 个 NAC 的子序列。如果无法做到这一点，则输出 -1。输入字符串中的任何大写字母都必须保留在其位置上。任何给定的测试用例都可能有多种可能的解决方案；任何正确的解决方案都将被接受。

实例

标准输入	标准输出
22 2 N??A?????C??????????	不是另一种建设性
18 0 卫星计数	卫星计数
2 1 ??	-1



问题 J. 护照印章

日

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：2048 兆字节

您刚刚拿到新护照，里面的页数已经准备好让移民官盖章。遗憾的是，由于您的护照页数太多，移民官懒得去有效利用您的护照页，因此您可能需要比您想象的更快换一本新护照.....

你准备了一些行程。每次旅行，当你通过护照检查时，移民官会找一些没有盖章的连续页，然后在所有页上盖章。因为移民官很懒，所以无法保证哪些连续页会盖章。

您将按顺序开始这些旅行，直到您的护照不再有足够的连续空页来满足下一次旅行的需要，这时您将申请一本新护照。您的计划是固定的--您不会中途跳过某次旅行，即使这意味着您今后可以进行更多的旅行。

如果移民官串通一气要对付你，那就找出你有可能因为护照没有足够的连续空白页而无法旅行的第一次旅行（或者，如果你总能在不缺空白页的情况下完成所有旅行）。

输入

第一行输入包含两个整数 n ($1 \leq n \leq 10^5$) 和 p ($1 \leq p \leq 10^{18}$)，其中 n 是您计划的旅行次数， p 是您新护照的页数。

接下来的 n 行中，每一行都包含一个整数 c ($1 \leq c \leq 10^{18}$)，即该行程需要盖章的连续页数。

输出

输出一个整数，这是在你需要一本新护照之前，由于没有足够的连续空白页供下一次旅行使用而可能需要的最少旅行次数。如果即使在最坏的情况下也能完成所有旅行，则输出 n 。

示例

标准输入	标准输出
5 15 1 2 3 4 5	3



问题 K. 阴影线

日

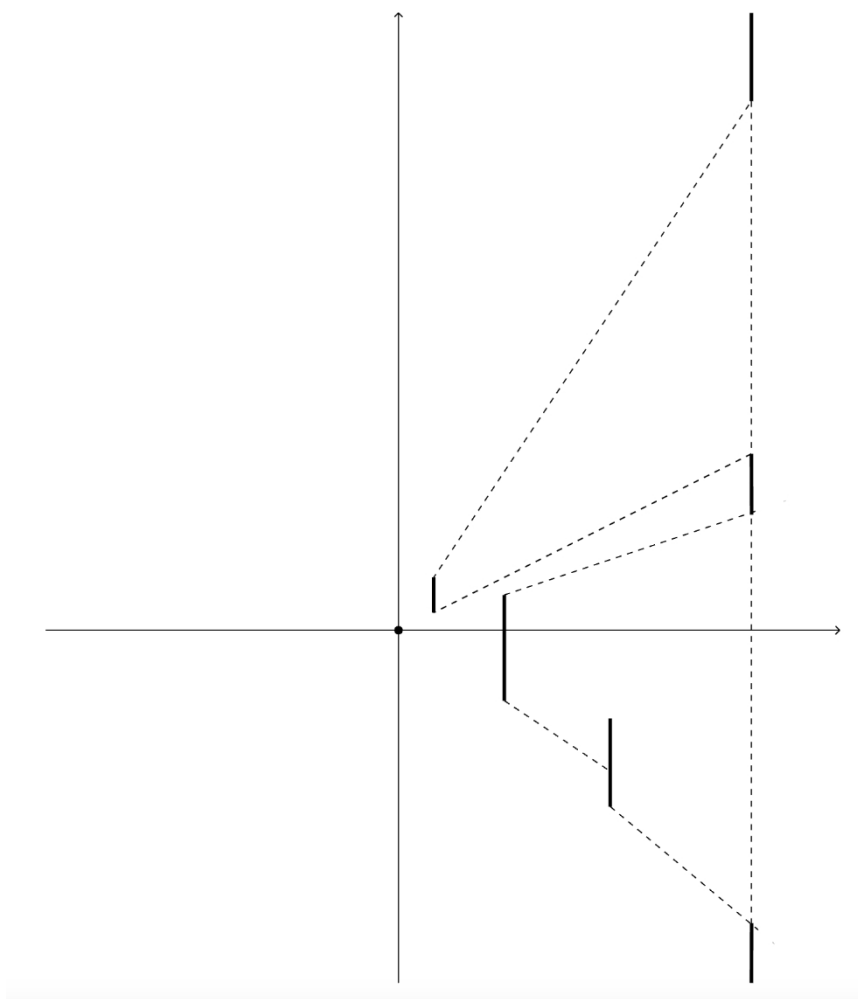
输入文件：标准输入

输出文件：标准输出

时间限制 10 秒 内存限制

2048 兆字节

在二维平面的原点有一个点光源。右边有一堵无限高的墙。光源和墙壁之间有一些不透明的垂直线段。因此，每条线段都会在墙上投下一个阴影。所有这些阴影重叠在一起，在墙上形成一个或多个间隔。



现在想象一下，将光源沿 x 轴向负方向移动，实际上是将光线拉得更远，使其远离直线上的所有物体。随着光源的移动，阴影也会移动，从而可能改变墙上阴影间隔的数量。你的任务是计算光源在墙壁上形成单个阴影区间的沿 x 轴

输入

第一行输入包含两个整数 n ($1 \leq n \leq 3\,000$) 和 w ($2 \leq w \leq 10^6$)，其中 n 是不透明垂直段的



数量， w 是无限高墙的 x 坐标。

日

接下来的 n 行中，每一行都包含三个整数 x ($0 < x < w$)、 y_{low} 和 y_{high} ($-10^6 \leq y_{low} < y_{high} \leq 10^6$)。每组三个整数描述了一条从 (x, y_{low}) 到 (x, y_{high}) 的线段。所有 y 坐标都是唯一的。没有两条线段相交或重叠。



输出

日

输出一个数字，即光源在墙壁上形成单个阴影区间的沿 x 轴的区间长度之和。如果该和包括一个无界区间（即光源无限远时有一个阴影区间），则打印-1。如果绝对误差或相对误差与评委答案的误差在 10^{-6} 以内，您的答案将被接受。

示例

标准输入	标准输出
3 20 2 1 3 6 -4 2 12 -10 -5	16.0



问题 L. 三角形的正方形

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：2048 兆字节

给你四个三角形边长的平方。请判断是否可以将它们（通过平移、旋转和反射）排列成一个正方形。任何三角形都不能重叠，也不能有空隙或漏洞。

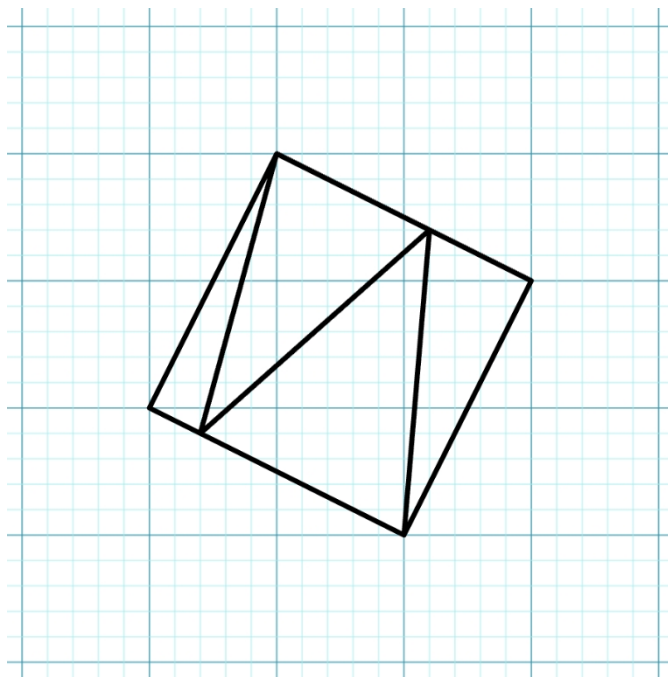


Рис. 4: 样本输入中第三个测试案例的解决方案。

输入

第一行输入包含一个整数 t ($1 \leq t \leq 20$)，即测试用例数。

接下来的 $4 \cdot t$ 行分别描述 t 个测试案例，每个案例由四个三角形组成，每行一个三角形。每个三角形由三个整数 a 、 b 和 c 组成 ($1 \leq a, b, c \leq 10^7$)。每个整数都等于三角形边长的平方。例如，如果三角形三条边的长度分别为 3、4 和 5，则输入值为 9 16 25。整数不一定是完全平方。但可以保证的是，所给的每个三元组都代表一个正面积的三角形。

输出

输出 t 行。按顺序为每个测试用例输出一行，每行一个整数，如果测试用例中的四个三角形可以排列成一个正方形，则该整数为 1，否则为 0。



示例

日

标准输入	标准输出
3	1
1 1 2	0
2 1 1	1
2 1 1	
1 2 1	
1 1 1	
1 1 1	
1 1 1	
1 1 1	
5 125 130	
125 20 145	
45 130 145	
145 145 80	



问题 M. 培训，第二轮

输入文件：标准输入
输出文件：标准输出
时间限制：6 秒 内存限制：2048 兆字节

阿什利正在布兰登的在线法官上为另一场编程比赛进行训练。布兰登的在线裁判仍然具有相同的功能，允许阿什利的教练汤姆载入问题列表，供阿什利练习。

汤姆为阿什利准备了一些问题。每个问题都有 "实施难度 "和 "思考难度 "两个参数，"实施难度 "有一个范围，"思考难度 "也有一个范围。

阿什利一开始有一定的执行技能和思考技能水平。阿什利将按照以下方式对汤姆策划的问题列表进行训练：她将查看列表中的第一个问题，然后解决该问题或跳过该问题。然后，她将按照汤姆加载问题的顺序，对列表中的每个问题重复上述步骤。一旦跳过一个问题，她就永远不能再回到这个问题。只有当阿什利当前的执行技能和当前的思考技能都在给定问题的范围内（含范围）时，她才会去解决问题。解决问题后，阿什利会对自己的经验进行反思，这样她就可以将执行技能提高一个等级，或将思考技能提高一个等级。她不能同时提高这两种技能，也不能跳过反思。

计算如果艾希莉最优化地计划她的思考，她最多可以解决多少个问题。

输入

第一行输入包含三个整数 n ($1 \leq n \leq 5\,000$)、 i 和 t ($0 \leq i, t \leq 10^9$)，其中 n 是汤姆给阿什利提出的问题数量， i 是阿什利的起始执行技能水平， t 是阿什利的起始思维技能水平。

接下来的 n 行中，每一行都包含四个整数 i_{low}, i_{high} ($0 \leq i_{low} \leq i_{high} \leq 2 \cdot 10^9$), t_{low}, t_{high} ($0 \leq t_{low} \leq t_{high} \leq 2 \cdot 10^9$)。每行描述一个问题，其执行难度范围为 $[i_{low}, i_{high}]$ ，其思考难度范围为 $[t_{low}, t_{high}]$ ，均包含在内。只有当阿什利当前的执行技能水平在执行难度范围内，且她的思维技能水平在思维难度范围内时，她才能解决问题。

输出

输出一个整数，即如果阿什利在每次解题后都以最佳方式选择反射，那么她能解决的最大问题数。

示例

标准输入	标准输出
3 0 0 0 1 0 1	3



1 1 0 1	日
1 1 1 1	