

CSP-S 2020 模拟赛

第二试

时间：2020 年 10 月 dd 日 hh:mm ~ hh:mm

AHSOFNU

PinkRabbit

题目名称	朝比奈实玖瑠的采购	长门有希的序列	古泉一树的游戏
题目类型	传统型	传统型	传统型
输入文件名	asahina.in	nagato.in	koizumi.in
输出文件名	asahina.out	nagato.out	koizumi.out
每个测试点时限	1.0 秒	3.0 秒	1.0 秒
内存限制	512 MB	512 MB	512 MB
测试点数目	25	10	25
测试点是否等分	是	是	是

提交源程序文件名

对于 C++ 语言	asahina.cpp	nagato.cpp	koizumi.cpp
-----------	-------------	------------	-------------

编译选项

对于 C++ 语言	-lm -O2 -std=c++11
-----------	--------------------

注意事项

1. 文件名（包括程序名和输入输出文件名）必须使用英文小写。
2. C++ 中函数 `main()` 的返回值类型必须是 `int`，值必须为 0。
3. 若无特殊说明，输入文件中同一行内的多个整数、浮点数、字符串等均使用一个空格分隔。
4. 若无特殊说明，结果比较方式为忽略行末空格、文末回车后的全文比较。
5. 原则上，每个测试点时限应为标准程序在该测试点上的运行时间的 2 倍及以上。
6. 每道题的时间限制、编译命令、是否开启文件输入输出等信息，在赛时均有可能变动，请各位选手以赛时通知为准。

朝比奈实玖瑠的采购 (asahina)

【题目背景】

「您好。」朝比奈学姐可爱地打招呼，店里大叔的脸瞬间就像是加热的天然沥青一样融化开来。

「你好，感谢你再度光临。」看来学姐已成了这家店的老主顾，和老板混得很熟了。

「嗯……要买哪一种好呢？」朝比奈学姐自言自语，以认真的眼神目不转睛审视写有定价和茶叶名称的手写海报。

想当然尔，我的茶水知识没有朝比奈学姐好，因此也无法给她建议，只好在她身旁闻着各种闻不惯的香味，弄得鼻子直发痒。

一涉及茶叶，态度就变严肃的朝比奈学姐，和大叔热心讨论着干燥的次数怎样又怎样、烘培的时机如何又如何。我就像是收割后的稻草人无所事事地枯站着。

「难得出来一趟——」朝比奈以比往常还要客气的神情仰望着我。

「阿虚，要不要来喝茶？这里的糯米丸子很好吃喔。也可以请店家帮忙冲泡刚买的茶叶……」

此时有件事，让我蛮挂念的。

朝比奈学姐似乎很在意时间。眼睛不时看着手表，心神似乎很不定。但是她的动作又很自然，不像是故意做给我看的，反倒像是很怕我察觉，不过很抱歉，我还是看出来了。因为学姐不只看了好几次时间，还不时在叹气。说她没有心事就太扯了。

「这丸子真好吃，茶也很香。不愧是朝比奈学姐千挑万选的茶叶。哎呀，真的很好喝。」尽管如此，我还是假装没发现学姐的心神不宁。对于体贴至此的自己，我都忍不住想夸奖自己了。

「嗯……」朝比奈学姐吃丸子吃得两颊鼓鼓的，缓缓低下头，又在看手表。

我突然有种预感，山雨欲来风满楼的预感。

……

「阿虚，你能帮我个忙吗？」朝比奈学姐恳求似的看着我。

学姐的表情相当真挚，我无意识地点了点头。

「春日让我采购一些茶叶为下次校庆做准备，卖茶赚的钱都算进文艺社的活动资金里，但是我不知道该怎么买才好。因为本金用的也是活动资金，我想要尽量帮长门多赚点钱。你帮我算一算怎么安排才能获得最多的活动资金好吗？好不好？虚君……」

又是恳求的眼神。用那种会让人化成果冻的表情和声音拜托，我根本没有招架的余地。

朝比奈学姐绽开了彩霞般的微笑。

「给你这个，我觉得在校庆的时候有些茶能卖个好价钱，都写在纸上了。谢谢你啦，阿虚。」

话说回来，校庆时的那些茶哪儿值那么多钱啊。那些肯出高价品茶的，在意的都是学姐你吧。

【题目描述】

茶叶店里有 n 个盒装的茶叶，学姐可以购买其中的一部分。每一盒当然只能买一次。

对于第 i 个盒装的茶叶，其中含有 c_i 包茶叶，每包茶叶的清香度为 w_i ，这一盒的价格为 v_i 。

学姐认为在校庆时有 m 种茶水的制作方法能卖个好价钱。这些方法也不能重复使用。

第 j 种方法需要 C_j 包茶叶，每包茶叶的清香度都应至少为 W_j ，制作后可以增加 V_j 的收入。

学姐想要购买 n 盒茶叶其中的一部分，然后在校庆时选择一些制作方法来获得收入。

请你求出学姐能获得的利润的最大值，也即最大化卖茶水带来的收入与购买茶叶的支出的差。

【输入格式】

从文件 *asahina.in* 中读入数据。

第一行，两个正整数 n, m ，分别表示盒装茶叶的数量以及茶水制作方法的数量。

接下来 n 行，第 i 行三个正整数 c_i, w_i, v_i ，表示第 i 盒茶叶包含的包数、清香度与价格。

接下来 m 行，第 j 行三个正整数 C_j, W_j, V_j ，表示第 j 种方法要求的包数、清香度与收入。

【输出格式】

输出到文件 *asahina.out* 中。

输出仅一行，一个数表示答案，即能获得的利润的最大值。

【样例 1 输入】

```
4 3
2 1800 10
20 2550 9999
4 2200 700
4 2000 750
6 1900 1500
3 2400 4550
1 1500 300
```

【样例 1 输出】

```
350
```

【样例 2】

见选手目录下的 *asahina/asahina2.in* 与 *asahina/asahina2.ans*。

【数据范围与提示】

对于所有测试点： $1 \leq n, m \leq 2000$ ， $1 \leq c_i, C_j \leq 50$ ， $1 \leq w_i, W_j, v_i, V_j \leq 10^9$ 。

每个测试点的具体限制见下表：

测试点编号	$n \leq$	$m \leq$	特殊限制
1 ~ 4	15	2000	无
5 ~ 8	2000	15	
9 ~ 12	100		$c_i = C_j = 1$
13 ~ 16	2000		$w_i = W_j = 1$
17 ~ 20			$v_i = V_j = 1$
21 ~ 25			无

长门有希的序列 (nagato)

【题目背景】

「他一见钟情的对象，并不是我。」

语调平稳得像是在念论文。

「他看到的我并不是我，而是资讯统合思念体。」

我静静聆听。长门又以同样的语调继续说明。

「只是他不晓得他看到的是什么。毕竟人类只是有机生命体，在意识层面上和资讯统合思念体是天差地远。恐怕他看到的是那超越时空的智慧与日积月累的知识吧。尽管他透过终端机读取到的资讯仅有九牛一毛，但那资讯带来的压力已足以令他为之倾倒。」

所以他才会会错意……吗？我看着长门参差不齐的刘海，叹了一口气。中河感受到的长门内在，事实上只是资讯统合思念体的某一端。虽然我不是很清楚，但是长门的确实拥有人类无法比拟的庞大历史、知识量等奇妙的力量。一不小心误闯进去的中河，为何会茫然自失一点也不足为奇了。

.....

「我还是有点好奇，你到底是怎么消除他拥有的能力的呢？」

尽管清楚那是人类所无法理解的知识，我仍然不自量力地提问。也许我也像中河那样抓狂了吧。

「那并不复杂，我可以告诉你，也许你可以理解。」

你是认真的？我有点不敢相信自己的耳朵。

「当然。他的超感应能力归根结底还是因为三年前凉宫春日造成的资讯爆炸，所以他的能力在我们看来也是以资讯的形式存在的。在人类这样的有机生命体中，受制于形式这些资讯只能以链状的形式彼此相连。而与资讯统合思念体交互时，总是链中的一段正在发挥作用。要么是资讯统合思念体让这些资讯全部变成它们的平方，要么是这些资讯将它们的总和发送给资讯统合思念体……」

长门的嘴一张一合，平稳的语调中传达出大量的信息，话语中提到的链状结构总让我回想起生物课上学到的 RNA。春日在生物课上常常发呆，但是她的成绩却比我高得多。

但果然还是听不懂啊，我让长门不用继续说下去了，这些信息只能使我的脑子拧成一团乱麻。

长门低下了头，数秒后又抬起头注视着我，神情中多了一丝落寞，她的眉毛下降了几毫米，那是只有我才能察觉到的差异。

凛冽的夜风吹得我耳畔刺痛，颤抖的双腿不禁后退了半步，但视野中长门的身影却突然潜入黑暗，像是快要溶解进虚空中的幽魂。

【题目描述】

请你维护一个长度为 n 的整数序列 $[a_1, a_2, \dots, a_n]$ ，执行 q 次操作，有如下两种操作：

- 1 1 r: 将区间 $[l, r]$ 中的每个元素变为自身的平方。即对每个 $l \leq i \leq r$ ，执行 $a_i \leftarrow a_i^2$ 。
- 2 2 l r: 询问区间 $[l, r]$ 中的所有元素的和，对 $p = 998244353$ 取模。即输出 $\left(\sum_{i=l}^r a_i \right) \bmod p$ 。

【输入格式】

从文件 *nagato.in* 中读入数据。

第一行，两个正整数 n, q ，分别表示序列长度和操作次数。

第二行， n 个正整数 a_1, a_2, \dots, a_n ，表示初始序列。

接下来 q 行，每行三个正整数 op, l, r ，依次表示每个操作。

【输出格式】

输出到文件 *nagato.out* 中。

对于每个 2 操作，输出一行一个数表示询问的答案。

【样例 1 输入】

```
3 3
1 2 3
2 1 3
1 1 2
2 2 3
```

【样例 1 输出】

```
6
7
```

【样例 2】

见选手目录下的 *nagato/nagato2.in* 与 *nagato/nagato2.ans*。

【数据范围与提示】

对于所有测试点： $1 \leq n, q \leq 2 \times 10^5$ ， $1 \leq a_i < p$ ， $op \in \{1, 2\}$ ， $1 \leq l \leq r \leq n$ 。

其中 $p = 998244353$ 。

每个测试点的具体限制见下表：

测试点编号	$n, q \leq$	特殊限制
1	5000	无
2 ~ 3	2×10^5	对于 1 操作有 $l = r$
4 ~ 5		对于 2 操作有 $l = r$
6 ~ 10		无

古泉一树的游戏 (koizumi)

【题目背景】

转过拐角，走廊的尽头的那间教室就是我每天放学后的去处。

那是文艺社活动室，虽说这间活动室已经被 SOS 团专横的团长据为己有。

从门边固定在墙上的铭牌就能看出这点：用图钉钉着、以潦草的字迹写着「With SOS 团」的字条安静地悬挂在那儿。

「欢迎，今天好像没有别人呢。」总是笑眯眯的古泉转过头，对刚开门的我这么说道。

真是受够了你那张一成不变的总感觉有点假笑的脸了啊。不过，你这是在做什么？

古泉似乎对桌上的奥赛罗棋盘上的巧克力板很在意。

「啊，这块巧克力是上次在鹤屋家的山上寻宝的时候挖出来的，更有趣的是每一小块巧克力正好和奥赛罗棋盘上的格子一样大。」

所以说这块巧克力板应该也属于女生们送给古泉的礼物的一部分吧。至于为什么能和奥赛罗扯上关系，我想那只是个巧合。

「如您所见，我没有把巧克力吃掉，不仅是因为我现在不饿，还因为我想在它上面玩个游戏。」

提到游戏，我和古泉之间的奥赛罗大战，古泉似乎就没赢过，也许这就是我在这方面比他强的超能力吧。

我饶有兴趣地问是什么游戏。

「你和我轮流从这块巧克力板上掰下一部分的巧克力吃掉，每次必须要吃掉至少一块巧克力，谁把巧克力全部吃完了就输掉游戏。」

谁没有礼貌地把东西全吃完，谁就输了啊。我回想起在家中吃饭时没有人想去夹走的最后一块鸡块，虽然最后还是被妹妹吃掉了。

「没错，很有生活气息吧。你能想到该怎么赢得游戏吗？」

那还不简单，我把所有巧克力全部吃光，只留下最后一块，你不就必须把它吃掉了么？

「很粗暴无礼的策略呢。」古泉再次微笑看着我。我只是在合理地利用规则嘛。

「真是抱歉，我没有解释清楚规则，巧克力固定在棋盘的左上角，每次应选择一小块巧克力，然后把它右下角的部分掰下。」

古泉补充道：「也就是掰下来的部分放回原有的位置时，必须是一个完整的右下角的形式。」

我好像理解了，这样一来我就无法一次性把所有的巧克力吃掉，只留最后一块了。因为最左上角的巧克力右侧和下侧都有巧克力啊。

「是的，这样一来游戏就没有那么简单了吧。在你来这里半个小时前，我想到了这样的游戏，要不要来玩玩看？」

.....

在下了几局之后，我的兴趣已经被消磨殆尽了，因为这几局都输给了古泉。不过古泉盯着桌面微笑着，好像还挺想继续下去的。

顺带一提，因为巧克力板只有一块，所以我们其实是用纸和笔模拟巧克力进行游戏的，所以我一块巧克力也没有下肚。

要是长门在这儿，估计她早就看透了古泉在这游戏中下的套了吧。既不能获胜，也不能吃巧克力的游戏，已经不太想玩下去了。

【题目描述】

给定一个组合游戏，状态用 $n \times m$ 的 01 矩阵描述。位置 (i, j) 为 1 就意味着这块巧克力还没有被吃掉。

每个合法的状态对应一个长度为 n 的整数序列 $[a_1, a_2, \dots, a_n]$ 。这个整数序列必须满足 $a_i \geq a_{i+1}$, $a_n \geq 0$ 且 $a_1 \leq m$ 。由此确定的矩阵 A 满足 $A_{i,j} = 1$ 当且仅当 $j \leq a_i$ 。

即 a_i 表示第 i 行还剩余的巧克力的个数，下一行的巧克力个数不多于上一行，第一行的巧克力个数不多于 m 。

状态转移通过在一个状态 A 中选取一个 $A_{i,j} = 1$ 的位置 (i, j) 进行：

- 令 $B = A$ 。
- 对于所有满足 $i \leq x \leq n$ 且 $j \leq y \leq m$ 的 (x, y) ，令 $B_{x,y} = 0$ 。
- 则此时 A 可以转移到 B 。

如果一名玩家操作后的状态为全 0 矩阵，则该玩家输掉游戏，对手赢得游戏。

假设对手聪明绝顶，己方先手，给定一个初始状态，请你判断该状态是否是己方的必胜态。

如果是必胜态，请给出下一步的最优决策，即下一步留给对手的必须是一个对手的必败态。

如果是必败态，请报告之。

【输入格式】

从文件 `koizumi.in` 中读入数据。

本题输入文件包含多组数据。

第一行，一个正整数 T ，表示数据组数。对于每组数据：

仅一行， $n + 1$ 个正整数 n, a_1, a_2, \dots, a_n ，表示初始状态对应的 a 序列。

【输出格式】

输出到文件 `koizumi.out` 中。

对于每组数据，如果初始状态是必胜态，输出 n 个数，表示下一步的最优决策对应的 a 序列，否则输出 Lose。

本题使用自定义校验器检验你的答案是否正确，因此若有多种满足条件的方案，你只需要输出任意一种。

【样例输入】

```
8
1 1
1 7
2 7 7
2 6 3
3 1 1 1
3 9 8 1
3 17 17 17
3 50 38 19
```

【样例输出】

```
Lose
1
7 6
4 3
1 0 0
2 2 1
17 17 12
Lose
```

【数据范围与提示】

对于所有测试点：

$1 \leq T \leq 1000$, $1 \leq n \leq 3$

$1 \leq a_1, a_2 \leq 10^9$ (如果 a_2 存在)

$1 \leq a_3 \leq 1000$ (如果 a_3 存在)

保证输入的是一个合法的状态，且至少有一种状态转移方式，即 $a_i \geq a_{i+1}$ 且 $a_1 \geq 1$ 。

每个测试点的具体限制见下表：

测试点编号	$n \leq$	特殊限制
1	1	无
2	2	$a_1 \leq 9$
3		$a_1 \leq 10^3$
4		无
5 ~ 7	3	$a_3 = 1$
8 ~ 10		$a_3 \leq 3$
11 ~ 13		$a_1 \leq 80$
14 ~ 18		$a_3 \leq 80$
19 ~ 23		$a_3 \leq 150$
24 ~ 25		无