



2021 南京信息工程大学 新生红包赛

Date: Oct 6, 2021

Host: ThinkSpirit

目录

本次赛事命题、奖项及奖品均由 ThinkSpirit 团队提供,最终解释权归 ThinkSpirit 团队所有。本试题册及试题仅供 2021 NUIST 新生红包赛使用, 未经 ThinkSpirit 团队允许禁止用于其他用途。本次试题中所提及的个人、团体及企业等与实际存在的个人、团体及企业均无关。

Problem A 欢迎来到提瓦特

Input file: standard input

Output file: standard output

Time limit: 1000ms

Memory limit: 512MB

题目描述

这里是七种元素交汇的幻想世界「提瓦特」。在遥远的过去，人们藉由对神灵的信仰，获赐了驱动元素的力量，得以在荒野中筑起家园。五百年前，古国的覆灭却使得天地变异…如今，席卷大陆的灾难已经停息，和平却仍未如期光临。

请输出一句话 `"Welcome to Tivat"`（不包括引号）。

输入描述

无任何输入。

输出描述

输出字符串一行字符串 `"Welcome to Tivat"`（不包括引号）。

测试样例

Standard Input	Standard Output
	Welcome to Tivat

Problem B 做饭

Input file: standard input

Output file: standard output

Time limit: 1000ms

Memory limit: 512MB

题目描述

雷电将军正在给旅行者做饭。因为旅行者着急去探险，所以雷电将军只能给旅行者做一个荤菜和一个素菜。设荤菜的美味度是 a ，素菜的美味度是 b ，旅行者所得到的美味度是荤菜美味度和素菜美味度的总和。

俗话说“人是铁饭是钢，一顿不吃饿得慌”。如果旅行者吃不饱饭，他所感受到的美味度会大大降低。准确的，如果雷电将军做荤菜或者素菜的份量是原来的 $\frac{1}{x}$ ($x \geq 1$) 那么他感受到的美味度是 $\lfloor (\frac{1}{x})^2 * a \rfloor$ 或者 $\lfloor (\frac{1}{x})^2 * b \rfloor$ 。然而，雷电将军并不擅长做饭。如果雷电将军把荤菜或者素菜烧糊了，那么这道菜的美味度是 $\lfloor -(\frac{1}{x})^2 * a \rfloor$ 或者 $\lfloor -(\frac{1}{x})^2 * b \rfloor$ 。

举个例子，假设 $a = 100, b = 50$ 。若雷电将军只做了 $\frac{1}{2}$ 分量的荤菜，则旅行者获得的美味度是 $100 * (\frac{1}{4}) + 50 = 75$ ，倘若雷电将军一不小心把这 $\frac{1}{2}$ 份量的荤菜做糊了，那么美味度则是 $-100 * (\frac{1}{4}) + 50 = 25$ 。

雷电将军可能会把饭做糊，也可能会缺斤少两，但她想尽可能提高两道菜的美味度。请聪明的你帮她算一算在最好的情况下美味度最大是多少。

注： $\lfloor x \rfloor$ 的意思是把 x 向下取整，例如 $\lfloor 15.8 \rfloor = 15$ 。

输入描述

第一行输入两个正整数 a, b ($1 \leq a, b \leq 10^9$)，中间用空格隔开。

输出描述

输出一个整数，表示最高的美味度。

测试样例

Standard Input	Standard Output
114514 1919810	2034324

Problem C 提瓦特炼金术

Input file: standard input

Output file: standard output

Time limit: 1000ms

Memory limit: 512MB

题目描述

“地水火风，领我救命。再造四大，重塑五行。”

wzy 从阿贝多那学到了神奇的炼金术，其原理如下：

$$Fe + Cr == Fr + Ce$$
$$Ce + Ru == Cu + Re$$
$$Cu + As == Au + Cs$$
$$Au + Lr == Ar + Lu$$
$$Ar + Fu == Au + Fr$$
$$Fr + Ce == Fe + Cr$$

通过这些方程式，他就可以将铁炼制成铜，将铜炼制成金，将金炼制成铁。

但由于 wzy 的炼金技术有限，他只能将 a 克的铁炼制成 b 克的铜，将 c 克的铜炼制成 d 克的黄金，将 e 克的黄金炼制成 f 克的铁。现在他想知道他能否通过有限的铁炼制出无限的黄金 (可以视为最初只有有限质量的铁，没有铜和金)。

wzy 当然知道解，但他还要去蒙德城门口喂鸽子，所以他把这个问题交给你了。

输入描述

输入有多行，第一行包含一个正整数 T ，接下来共有 T 行，每行包含 6 个整数 a, b, c, d, e, f ， $0 < a, b, c, d, e, f \leq 10^6$ 。

输出描述

输出共 T 行，每行表示一组输入的答案，每组输出占一行。如果旅行者可以通过有限质量的铁创造出无限质量的金时，输出 "Awesome!"，否则输出 "Stupid!" (不包括引号)。

测试样例

Standard Input	Standard Output
2	Awesome!
100 200 250 150 200 250	Stupid!
1 1 1 1 10 1	

Problem D 提瓦特初等元素论

Input file: standard input

Output file: standard output

Time limit: 1000ms

Memory limit: 512MB

题目描述

众所周知，在提瓦特大陆上除去草元素 (Dendro) 外有 6 种元素，他们分别是：火 (Pyro)、水 (Hydro)、风 (Anemo)、雷 (Electro)、冰 (Cryo)、岩 (Geo)。他们之间两两会发生反应：

水和冰相遇发生冻结反应 (Frozen)

火和水相遇会发生蒸发反应 (Vaporize)

水和雷相遇会发生感电反应 (Electro-Charged)

火和雷相遇会发生超载反应 (Overloaded)

冰和雷相遇会发生超导反应 (Superconduct)

冰和火相遇会发生融化反应 (Melt)

风元素和除岩元素外的其他元素会发生扩散反应 (Swirl)

岩元素和除风元素外的其他元素会发生结晶反应 (Crystallize).

提瓦特大陆上元素攻击是一种强力的攻击方式，因此东走夕对元素反应非常感兴趣。他发现提瓦特大陆上的元素攻击不但元素类型不同，其中包含的**元素量**也不同。为了深入研究提瓦特元素论，东走夕找了一只落单的丘丘王，并对它施加了各种各样的元素攻击。东走夕在暴打丘丘王时发现了以下规律：

1、风元素和岩元素不会产生元素附着。

2、如果丘丘王身上没有元素附着，那么元素攻击会让丘丘王带上和此次元素攻击同类型且等元素量的元素。

3、如果丘丘王身上已经有元素附着，那么不同类型元素攻击会在丘丘王的身上发生剧烈的反应。反应时会消耗丘丘王身上的元素附着量，值等于此次元素攻击的元素量。但如果此次攻击的元素量大于丘丘王身上的元素附着量，只会消耗完元素附着量但不会产生新的元素附着。

4、如果丘丘王身上已经有元素附着，那么同类型元素攻击会刷新丘丘王身上的元素附着，新的元素附着量为元素攻击和已有元素附着量的较大值。

东走夕被元素反应深深吸引，以至于忘记了记录发生了什么反应，所幸他记得自己释放的每一次元素攻击的类型和元素量。现在，东走夕想请你告诉他发生了什么反应。

输入描述

第一行一个正整数 n 。

接下来 n 行，每行包括一个字符串 s_i 和正整数 a_i ，表示元素攻击的类型和元素量，中间用一个空格隔开。

输出描述

每行一个字符串，表示发生的反应类型。

测试样例

Standard Input	Standard Output
10 Electro 6 Cryo 9 Pyro 5 Anemo 1 Pyro 1 Pyro 4 Cryo 2 Anemo 1 Geo 6 Hydro 1	Superconduct Swirl Melt Swirl Crystallize

样例解释

第一次攻击，将丘丘王附上雷元素，元素附着量为 6。

第二次攻击，丘丘王身上的雷元素和冰元素反应，发生超导反应，此时丘丘王身上无元素附着。

第三次攻击，将丘丘王附上火元素，元素附着量为 5。

第四次攻击，丘丘王身上的火元素和风元素反应，发生扩散反应，此时丘丘王身上还剩余 4 元素量的火元素附着。

第五、六次攻击，刷新丘丘王身上的火元素附着，火元素附着量为 4。

第七次攻击，丘丘王身上的火元素和冰元素反应，发生融化反应，剩余 2 元素量的火元素附着。

第八次攻击，丘丘王身上的火元素和风元素反应，发生扩散反应，剩余 1 元素量的火元素附着。

第九次攻击，丘丘王身上的火元素和风元素反应，发生结晶反应，此时丘丘王身上无元素附着。

第十次攻击，将丘丘王附上水元素，元素附着量为 1。

数据规模

对于 20% 的数据: $1 \leq n \leq 10, 1 \leq a_i \leq 10$

对于 50% 的数据: $1 \leq n \leq 1000, 1 \leq a_i \leq 100$

对于 100% 的数据: $1 \leq n \leq 100000, 1 \leq a_i \leq 1000$

Problem E 共鸣

Input file: standard input

Output file: standard output

Time limit: 1000ms

Memory limit: 512MB

题目描述

在提瓦特世界中，存在着一种很特殊的元素现象-岩元素共鸣。岩元素共鸣通常出现在岩造物之间。

套餐用户在南天门的伏龙树下发现了一个奇怪的封印，一阵阴风袭来，套餐用户被吸入封印中。套餐用户来到了一个奇特的地界，在此地，岩造物不断地被创造以及消散。经过一段时间的观察，套餐用户发现其中只有两种岩造物，特殊的岩造物以及普通的岩造物。

特殊的岩造物可以与同时与多个普通的岩造物产生共鸣，同样的，普通岩造物可以同时与多个特殊岩造物产生共鸣。特殊的岩造物之间不能产生共鸣现象，同样的，普通的岩造物之间也不会产生共鸣现象。

岩造物与岩造物之间每秒产生一次共鸣，一个特殊岩造物与多个普通岩造物可以同时产生多次共鸣。特殊岩造物与普通岩造物之间的共鸣无视距离。岩造物刚产生的那一刻会产生共鸣，岩造物毁灭的那一刻不会产生共鸣。

在这个地界中，岩造物不断地生成或毁灭，没有一个岩造物会永久存在。套餐用户在这个地界中呆了很久，四处游行。他发现只需要从某一秒（这一秒没有产生岩元素共鸣）开始，到某一秒（这一秒恰好也没有产生岩元素共鸣）结束，数出这一段时间内总共产生了多少次岩元素共鸣并且将其刻在石板上，如果答案正确，即可离开此地。套餐用户分析完后就立刻开始计数，我们将开始时刻设置为时刻 1，即从时刻 1 开始考虑本问题。这对套餐用户来说似乎有点困难，套餐用户请你帮一帮他。

输入描述

第一行输入一个整数 $N(1 \leq N \leq 1,000,000)$ ，表示接下来将会有 N 个岩造物生成。

接下来的 N 行，每一行输入三个正整数 $s_i(1 \leq s_i \leq 1,100,000)$, $e_i(1 \leq e_i \leq 1,100,000)$, $t_i(t_i \in \{0,1\})$ ，分别表示第 i 个岩造物的被创造时间，毁灭时间，岩造物属性。其中， $t_i = 1$ 表示第 i 个岩造物为特殊岩造物， $t_i = 0$ 表示第 i 个岩造物为普通岩造物。保证 $e_i > s_i$ 。

输入数据保证必定有两个时刻，使得没有岩元素共鸣产生。

输出描述

输出从第一次出现没有岩元素共鸣产生的时刻，到第二次出现没有岩元素共鸣产生的时刻之间，总共产生了多少次岩元素共鸣。

由于答案可能很大，请输出结果模 19260817 后得出的答案。

测试样例

Standard Input	Standard Output
5 1 2 1 1 2 0 3 10 1 3 10 0 4 10 1	13

样例解释

有 5 个岩造物将被创造。

第一个岩造物在时刻 1 被创造出来，在时刻 2 毁灭。是特殊岩造物

第二个岩造物在时刻 1 被创造出来，在时刻 2 毁灭。是普通岩造物

.....

第五个岩造物在时刻 4 被创造出来，在时刻 10 毁灭。是特殊岩造物

根据数据可以得到每个时刻的特殊岩造物和普通岩造物的数量

时刻 1：特殊 1 个，普通 1 个

时刻 2：特殊 0 个，普通 0 个（因为在这个时刻毁灭了）[第一个时刻，使得没有共鸣产生]

时刻 3：特殊 1 个，普通一个，一次共鸣

时刻 4：特殊 2 个，普通一个，二次共鸣

时刻 5：同上

时刻 6：同上

时刻 7：同上

时刻 8：同上

时刻 9：同上

时刻 10: 特殊 0 个，普通 0 个 [第二个时刻，使得没有共鸣产生]

从时刻 2 开始到时刻 10 结束，一共产生 $1 + 2 * 6 = 13$ 次共鸣，所以输出 13。

提示

```
int res = 100;
int ress = res % 3; // 这个就是模运算，即取除3的余数
cout << ress; // 输出1
```


Problem F 稻妻灯谜

Input file: standard input

Output file: standard output

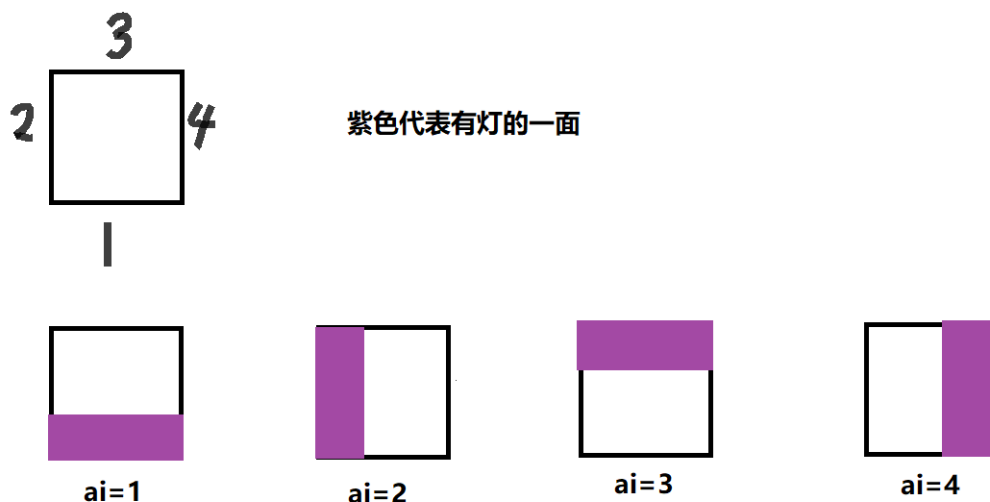
Time limit: 1000ms

Memory limit: 512MB

题目描述

在稻妻有一种灯谜游戏。有 n 个正四棱柱，每个正四棱柱有 4 个侧面，有一个旋转轴，旋转轴通过上底面的中心和下底面的中心，从上往下看可以绕旋转轴顺时针方向旋转。4 个侧面中有一个面有灯，我们称这个面是正面。一开始正面的朝向随机，可能全部朝向玩家，也可能朝向不同的方向。你的任务是顺时针转动这 n 个正四棱柱使得他们的正面全部朝向你。然而有些正四棱柱和其他柱子有连接，使得在顺时针转动某一个柱子一次，可能会有 $0 \sim n-1$ 个正四棱柱和那个柱子同时顺时针旋转一次。注意连接是单向的，若转动第 i 个柱子时第 j 个柱子会同时转动，当你转动第 j 个柱子时，第 i 个柱子不一定转动，反之同理。

图片是俯视图。为了方便表达，我们约定朝向玩家的方向（向下）为编号 1，依次顺时针标号其他三个方向。我们把这 n 个正四棱柱初始有灯的那一面朝向记为 a_1, a_2, \dots, a_n ，其中 $1 \leq a_i \leq 4$ ，有 n 阶 01 矩阵 $b_{1,1}, b_{1,2}, \dots, b_{i,j}, \dots, b_{n,n-1}, b_{n,n}$ ，若 $b_{i,j} = 1$ ，则表示顺时针旋转第 i 个柱子一次，第 j 个柱子也会同时顺时针旋转一次。若 $b_{i,j} = 0$ ，则表示顺时针旋转第 i 个柱子一次，第 j 个柱子不动。



你的任务是转动每个柱子 0 次或者若干次，使得正面全部朝向玩家（1 号面），输出任意一个可行解即可。

输入描述

第一行是一个整数 n ，表示正四棱柱的数量， $4 \leq n \leq 12$ 。

下一行是 n 个整数，表示 a_1, a_2, \dots, a_n ，其中 $1 \leq a_i \leq 4$ 。

第 3 行到第 $n + 2$ 行，每行有 n 个整数（0 或 1）。第 $i + 2$ 行的值表示 $b_{i,1}, b_{i,2}, \dots, b_{i,n}$ 的值， $b_{i,i}$ 总是为 1。

输入数据保证有解。

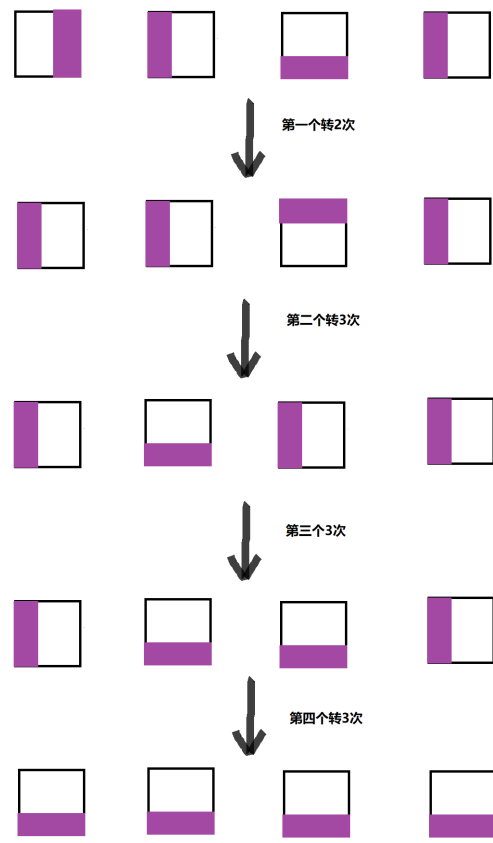
输出描述

一行 n 个整数，整数之间一个空格隔开，第 i 个整数表示顺时针转动第 i 个柱子多少次。允许的范围为 0 到 10^9 。如果存在多个解，请输出任意一个解。

测试样例

Standard Input	Standard Output
4 4 2 1 2 1 0 1 0 0 1 1 0 0 0 1 0 1 0 0 1	2 3 3 3

样例解释



Problem G 策划我爱你

Input file: standard input

Output file: standard output

Time limit: 1000ms

Memory limit: 512MB

题目描述

新版本更新后，东走夕发现新怪物相当强力（刻师傅刮不动了），新角色机制诡异（特指某只鱼），活动送的原石又是那么的少（根本不够）。于是，东走夕发自内心的喊出：“策划我爱你！”。

为了表达对策划的感激之情，东走夕写下了一个数学公式：
$$\sum_{n=a}^b \sum_{m=c}^d w C_m^n$$

由于东走夕的数学功底十分感人，所以他想麻烦你帮他输出 $\sum_{n=a}^b \sum_{m=c}^d w C_m^n$ 的值。

由于答案的值可能非常大所以东走夕希望答案对 998244353 取模。

输入描述

第一行一个整数 k 。

接下来 k 行，每行 5 个整数 a, b, c, d, w 。

输出描述

共 k 行，每行一个整数表示答案。

测试样例

Standard Input	Standard Output
3	
1 1 1 1 1	1 80
1 2 3 4 5	2950
2 4 6 8 10	

数据规模

对于 10% 的数据： $1 \leq k \leq 10, 1 \leq b \leq 10, 1 \leq d \leq 10, w \leq 1000$

对于 30% 的数据： $1 \leq k \leq 100, 1 \leq b \leq 100, 1 \leq d \leq 100, w \leq 100000$

对于 100% 的数据： $1 \leq k \leq 100000, 1 \leq b \leq 5000, 1 \leq d \leq 5000, w \leq 100000000$

数据保证 $1 \leq a \leq b \leq c \leq d$

Problem H 从蒙德到璃月

Input file: standard input

Output file: standard output

Time limit: 1000ms

Memory limit: 512MB

题目描述

“向着星辰与深渊！欢迎来到冒险家协会。”

蒙德冒险家协会的凯瑟琳让东走夕赶去璃月参加请仙典仪。可是蒙德和璃月之间的每一段道路都充满了怪物，消灭这些怪物东走夕会损失一定的生命值。由于芭芭拉小姐不愿意离开蒙德，东走夕没有恢复生命值的手段。但派蒙告诉东走夕，某些城镇中建有七天神像，东走夕可以在那里恢复满生命值。

应急食品派蒙告诉了东走夕城镇的分布，包括所有城镇之间的距离和建有七天神像城镇的位置。由于派蒙记性不好，派蒙给了每个城市一个独立的编号，这个编号从 1 到 n ，其中点 1 表示出发点蒙德， n 表示终点璃月。

而东走夕对自己的能力非常清楚，他明白自己在满生命值的情况下最多走 s 的距离 (如果东走夕在进入有七天神像的城市时刚好耗尽生命值则不会倒下)。

由于请仙典仪就要到了，东走夕希望知道在满生命值的情况下，安全的从蒙德出发到璃月的最短路程是多少。

输入描述

第一行四个整数 n, m, k, s 。

n 表示城镇数量， $m(1 \leq m \leq n(n-1)/2)$ 表示城镇之间的道路数量， k 表示拥有七天神像的城镇的数量， s 表示东走夕在不补充生命值情况下最多行走的距离。

第二行 k 个整数，其值介于 2 和 $n-1$ 之间，表示拥有七天神像的城镇。

接下来 m 行每行三个整数 a_i, b_i, w_i ，表示城镇 a_i 到城市 b_i 之间需要走过 w_i 的路程。其中 $1 \leq a_i, b_i \leq n$ 。保证两个城镇之间最多只有一条路直接相连。

输出描述

一个整数，输出从蒙德出发到璃月的最短路程 Dis ，保证 $Dis < 2^{31}$ 。

如果无法安全到达，输出 "Maybe I wasn't meant for the world..." (不包括引号。翻译：世界... 拒绝了我)。

测试样例

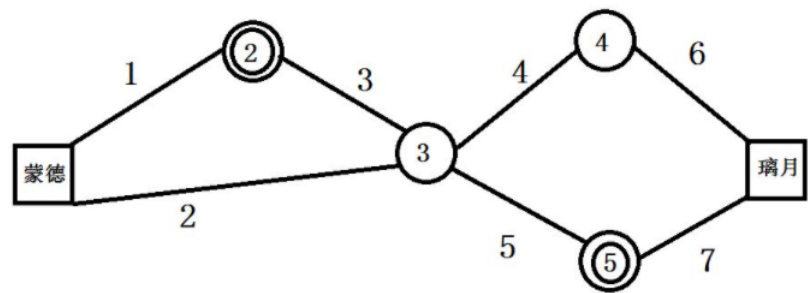
Standard Input	Standard Output
6 7 2 7 2 5 1 2 1 1 3 2 2 3 3 3 4 4 3 5 5 4 6 6 5 6 7	14
6 7 2 6 2 5 1 2 1 1 3 2 2 3 3 3 4 4 3 5 5 4 6 6 5 6 7	Maybe I wasn't meant for the world...

数据规模

对于 50% 的样例： $2 \leq n \leq 10, 1 \leq s < 10000, 1 \leq w_i < 1000$ 。
对于 100% 的样例： $2 \leq n \leq 500, 1 \leq s < 2^{31}, 1 \leq w_i < 2^{21}$ 。

样例解释

对于样例 1，东走夕走到点 3 时最多还可以在走 5 米，虽然 $3 \rightarrow 4 \rightarrow 6$ 在路程上更近，但东走夕会因为生命值耗尽倒下，所以只能选择下方 $3 \rightarrow 5 \rightarrow 6$ 的路径。



对于样例 2，东走夕无论怎么走都会倒下，所以输出 "Maybe I wasn't meant for the world ...".