

2019年华南理工大学软件学院“新生杯”

正式赛题

未正式开赛前，请不要翻阅本试题!

如有任何疑问，请及时联系在场工作人员。

华南理工大学软件学院ACM集训队

目录

序号	题目
1001	有趣的故事
1002	有趣的骨牌
1003	zzy和zbs的葡萄
1004	zbs和小布鞋的回文数
1005	zzy的数学水题
1006	有趣的括号匹配
1007	有趣的数组
1008	火箭测试
1009	au孙的奖牌
1010	有趣的AB串
1011	zzy爱下棋
1012	有趣的比赛
1013	有趣的签到题

1001 有趣的故事

Time limit: 1000ms Memory limit: 256Mb

Description

《三体》中有着这样一个关于计算机的有趣的故事。

"所以，伟大的皇帝，您的帝国还要发展!"冯·诺伊曼不失时机地说，"如果掌握了太阳运行的规律，你就能充分利用每一个恒纪元，同时避免乱纪元带来的损失，这样发展速度比西洋要快得多。请你相信我们，我们是学者，只要能用三定律和微积分准确预测太阳的运行，不在乎谁征服统治世界。"

"朕当然需要预测太阳的运行，但你们让我集结三千万大军，至少要首先向朕演示一下这种计算如何进行吧?"

"陛下，请给我三个士兵，我将为您演示。"冯·诺伊曼兴奋起来。

"三个?只要三个吗，朕可以轻易给你三千个。"秦始皇用不信任的目光扫视着冯·诺伊曼。

伟大的陛下，您刚提到东方人在科学思维上的缺陷，就是因为你们没有意识到，复杂的宇宙万物其实是由最简单的单元构成的。我只要三个，陛下。"

秦始皇挥手召来了三名士兵，他们都很年轻，与秦国的其他士兵一样，一举一动像听从命令的机器。

"我不知道你们的名字，"冯·诺伊曼拍拍前两个士兵的肩，"你们两个负责信号输入，就叫'入1'、'入2'吧，"他又指指最后一名士兵，"你，负责信号输出，就叫'出'吧，"他伸手拨动三名士兵，"这样，站成一个三角形，出是顶端，入1和入2是底边，"

"哼，你让他们成楔形攻击队形不就行了?"秦始皇轻蔑地看着冯·诺伊曼。

牛顿不知从什么地方掏出六面小旗，三白三黑，冯·诺伊曼接过来分给三名士兵，每人一白一黑，说:白色代表0，黑色代表1。好，现在听我说，出，你转身着着入1和入2，如果他们都举黑旗，你就举黑旗，其他的情况你都举白旗，这种情况有三种:入1白，入2黑;入1黑，入2白;入1、入2都是白。"

"我觉得你应该换个颜色，白旗代表投降。"秦始皇说。

兴奋中的冯·诺伊曼没有理睬皇帝，对三名士兵大声命令:"现在开始运行!入1入2，你们每人随意举旗，好，举!好，再举!举!"

入1和入2同时举了三次旗，第一次是黑黑，第二次是白黑，第三次是黑白。出都进行了正确反应，分别举起了一次黑和两次白。

"很好，运行正确，陛下，您的士兵很聪明!"

"这事儿傻瓜都会，你能告诉朕，他们在干什么吗?"秦始皇一脸困惑地问。

"这三个人组成了一个计算系统的部件，是门部件的一种，叫'与门'。"冯·诺伊曼说完停了一会儿，好让皇帝理解。

秦始皇面无表情地说："朕是够郁闷的，好，继续。"

冯·诺伊曼转向排成三角阵的三名士兵:"我们构建下一个部件。你，出，只要看到入1和入2中有一个人举黑旗，你就举黑旗，这种情况有三种组合黑黑、白黑、黑白，剩下的一种情况一一白白，你就举白旗。明白了吗?好孩子，你其聪明，门部件的正确运行你是关键，好好干，皇帝会奖赏你的!下面开始运行:举!好，再举!再举!好极了，运行正常，陛下，这个门部件叫或门。"

然后，冯·诺伊曼又用三名士兵构建了与非门、或非门、异或门、同或门和三态门，最后只用两名士兵构建了最简单的非门，出总是举与入颜色相反的旗。

冯·诺伊曼对皇帝鞠躬说："现在、陛下，所有的门部件都已演示完毕，这很简单不是吗？任何三名士兵经过一小时的训练就可以掌握。"

"他们不需要学更多的东西了吗？"秦始皇问。

"不需要，我们组建一千万个这样的门部件，再将这些部件组合成一个系统，这个系统就能进行我们所需要的运算，解出那些预测太阳运行的微分方程。这个系统，我们把它叫做.....嗯，叫做....."

"计算机。"汪淼说。

"啊——好！"冯·诺伊曼对汪淼竖起一根指头，"计算机，这个名字好，整个系统实际上就是一部庞大的机器，是有史以来最复杂的机器！"

游戏时间加快，三个月过去了。

秦始皇、牛顿、冯·诺伊曼和汪淼站在金字塔顶部的平台上，这个平台与汪淼和墨子相遇时的很相似，架设着大量的天文观测仪器，其中有一部分是欧洲近代的设备。在他们下方，三千万秦国军队宏伟的方阵铺展在大地上，这是一个边长六公里的正方形。在初升的太阳下，方阵凝固了似的纹丝不动，仿佛一张由三千万个兵马俑构成的巨毯，但飞翔的鸟群误入这巨毯上空时，立刻感到了下方浓重的杀气，鸟群顿时大乱，惊慌混乱地散开或绕行。汪淼在心里算了算，如果全人类站成这样一个方阵，面积也不过是上海浦东大小，比起它表现的力量，这方阵更显示了文明的脆弱。

"陛下，您的军队其是举世无双，这么短的时间，就完成了如此复杂的训练。"冯·诺伊曼对秦始皇赞叹道。

"虽然整体上复杂，但每个士兵要做的很简单，比起以前为粉碎马其顿方阵进行的训练来，这算不了什么。"秦始皇按着长剑剑柄说。

"上帝也保佑，连着两个这样长的恒纪元。"牛顿说。

"即使是乱纪元，朕的军队也照样训练，以后，他们也会在乱纪元完成你们的计算。"秦始皇骄傲地扫视着方阵说。

"那么，请陛下发出您伟大的号令吧！"冯·诺伊曼用激动得发颤的声音说。

秦始皇点点头，一名卫士奔跑过来，握住皇帝的剑柄向后退了几步，抽出了那柄皇帝本人无法抽出的青铜长剑，然后上前跪下将剑呈给皇帝，秦始皇对着长空扬起长剑，高声喊造：

"成计算机队列！"

金字塔四角的四尊青铜大鼎同时轰地燃烧起来，站满了金字塔面向方阵一面坡墙的士兵用宏大的合唱将始皇帝的号令传诵下去：

"成计算机队列！"

下面的大地上，方阵均匀的色彩开始出现扰动，复杂精细的回路结构浮现出来，并渐渐充满了整个方阵，十分钟后，大地上出现了一块三十六平方公里的计算机主板。

冯·诺伊曼指着下方巨大的人列回路开始介绍："陛下，我们把这台计算机命名为'秦一号'。请看，那里，中心部分，是CPU，是计算机的核心计算元件，由您最精锐的五个军团构成，对照这张图您可以看到里面的加法器、寄存器、堆栈存储器；外围整齐的部分是内存，构建这部分时我们发现人手不够，好在这部分每个单元的动作最简单，就训练每个士兵拿多种颜色的旗帜，组合起来后，一个人就能同时完成最初二十个人的操作，这就使内存容量达到了运行'秦1.0'操作系统的最低要求；你再看那条贯穿整个阵列的通道，还有那些在通道上待命的轻骑兵，那是BUS，系统总线，负责在整个系统间传递信息。"

"总线结构是个伟大的发明，新的插件，最大可由十个军团构成，能够快捷地挂接到总线上运行，这使得‘秦一号’的硬件扩展和升级十分便利；再看最远处那一边，可能要用望远镜才能看清，那是外存，我们又用了哥白尼起的名字，叫它‘硬盘’，那是由三百万名文化程度较高的人构成，您上次坑儒时把他们留下是对的，他们每个人手中都有一个记录本和笔，负责记录运算结果，当然，他们最大的工作量还是作为虚拟内存，存贮中间运算结果，运算速度的瓶颈就在他们那里。这儿，离我们最近的地方，是显示阵列，能显示计算机运行的主要状态参数。"

冯·诺伊曼和牛顿搬来一个一人多高的大纸卷，在秦始皇面前展开来，当纸卷展到尽头时，汪淼一阵头皮发紧，但他想象中的匕首并没有出现，面前只有一张写满符号的大纸，那些符号都是蝇头大小，密密麻麻，看上去与下面的计算机阵列一样令人头晕目眩。

"陛下，这是就我们开发的‘秦1.0’版操作系统，计算软件将在它上面运行。陛下您看"冯·诺伊曼指指下面的人列计算机，"这阵列是硬件，而这张纸上写的是软件，硬件和软件，就如同琴和乐谱的关系。"说着他和牛顿又展开了一张同样大小的纸，"陛下，这就是用数值法解那一组微分方程的软件，将天文观测得到的三个太阳在某一时间断面的运动矢量输入，它的运行就能为我们预测以后任一时刻太阳的运行状态。我们这次计算，将对以后两年太阳的运行做出完整预测，每组预测值的时间间隔为一百二十小时。"

秦始皇点点头："那就开始吧。"

冯·诺伊曼双手过顶，庄严地喊道："奉圣上御旨，计算机启动！系统自检！"

在金字塔的中部，一排旗手用旗语发出指令，一时间，下面大地上三千万人构成的巨型主板仿佛液化了，充满了细密的粼粼波光，那是几千万面小旗在挥动。在靠近金字塔底部的显示阵列中，一务由无数面绿色大旗构成的进度条在延伸着，标示着自检的进度。十分钟后，进度条走到了头。

"自检完成！引导程序运行！操作系统加载！！"

下面，贯穿人列计算机的系统总线上的轻骑兵快速运动起来，总线立刻变成了一条湍急的河流，这河流沿途又分成无数条细小的支流，渗入到各个模块阵列之中。很快，黑白旗的涟漪演化成汹涌的浪潮，激荡在整块主板上。中央的CPU区激荡最为剧烈，像一片燃烧的火药。突然，仿佛火药燃尽，CPU区的扰动渐渐平静下来，最后竟完全静止了，以它为圆心，这静止向各个方向飞快扩散开来，像快速封冻的海面，最后整块主板大部分静止了，其间只有一些零星的死循环在以不变的节奏没有生气地闪动着，显示阵列中出现了闪动的红色。

"系统锁死！"一名信号官高喊。故障原因很快查清，是CPU状态寄存器中的一个门电路运行出错。

"系统重新热启动！"冯·诺伊曼胸有成竹地命令道。

"慢！"牛顿挥手制止了信号官，转身一脸阴毒地对秦始皇说，"陛下，为了系统的稳定运行，对故障率较高的部件应核采取一些维修措施。"

秦始皇拄着长剑说："更换出错部件，组成那个部件的所有兵卒，斩！以后故障照此办理。"

冯·诺伊曼厌恶地看了牛顿一眼，看着一组利剑出鞘的骑兵冲进主板，"维修"了故障部件后，重新发布了热启动命令。这次启动十分顺利，二十分钟后，三体世界的冯·诺伊曼结构人列计算机在"秦1.0"操作系统下进入运行状态。

"启动太阳轨道计算软件'Tree-Body 1.0'！"牛顿声嘶力竭地发令，"启动计算主控！加载差分模块！加载有限元模块！加载谱方法模块……调入初始条件参数！计算启动！！"

主板上波光粼粼，显示阵列上的各色标志此起彼伏地闪动，人列计算机开始了漫长的计算。

"真是很有意思。"秦始皇手指壮观的计算机说，"每个人如此简单的行为，竟产生了如此复杂的大东西！欧洲人骂朕独裁暴政，扼杀了社会的创造力，其实在严格纪律约束下的大量的人，合为一个整体后也能产生伟大的智慧。"

"伟大的始皇帝，这是机器的机械运行，不是智慧。这些普通卑贱的人都是一个个0，只有在最前面加上您这样一个1，他们的整体才有意义。"牛顿带着奉承的微笑说。

"恶心的哲学。"冯·诺伊曼瞥了牛顿一眼说，"如果到时候，按你的理论和数学模型计算出的结果与预测不符，你我可就连零都不是了。"

"对，那时你们可其的什么都不是了!"秦始皇说着，拂袖而去。

时光飞逝，人列计算机运行了一年零四个月，除去程序的调试时间，实际计算时间约一年两个月，这期间，只因乱纪元过分恶劣的气候中断过两次，但计算机存贮了中断现场数据，都成功地从断点恢复了运行。当秦始皇和欧洲学者们再次登上金字塔顶部时，第一阶段的计算已经完成，这批结果数据，精确地描述了以后两年太阳运行的轨道状况。

这是一个寒冷的黎明，彻夜照耀着巨大主板的无数火炬已经熄灭，计算机完成后，'秦1.0'进入待机状态，主板表面汹涌的浪涛变成了平静的微波。

故事讲完了。

现在定义如下代号

1：冯·诺伊曼

2：秦始皇

3：牛顿

4：汪淼

给定代号，请你输出代号对应的人名在文章当中出现了几次。

Input Description

第一行一个整数 T 表示数据组数。 $(T \leq 10)$

接下来每一行一个整数 n 表示代号。 $(1 \leq n \leq 4)$

Output Description

对于每个代号，输出一个整数表示对应名字在文中出现了几次。

Sample Input

```
4
1
2
3
4
```

Sample Output

```
21
19
10
6
```

1002 有趣的骨牌

Time limit: 1000ms Memory limit: 256Mb

Description

这是一个有趣的骨牌游戏。

给定一个 $n * n$ 的棋盘，任意挖去两个格子，问剩下的空间是否能用 $1 * 2$ 的骨牌铺满。

骨牌可以任意旋转，并且能覆盖任意相邻的两个格子。骨牌之间不能重叠且不限制骨牌的使用数量。

如果可以输出“Yes”，否则输出“No”。

Input Description

第一行一个正整数 T ，表示数据组数。 $(1 \leq T \leq 1000)$

接下来 T 行，每行5个整数， $n, x1, y1, x2, y2$ 。 $(1 \leq n, x1, y1, x2, y2 \leq 1000, \text{保证} n \text{为偶数。})$

Output Description

对于每组测试用例，输出“Yes”或“No”且独占一行。

Sample Input

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

Sample Output

```
No
Yes
```

1003 zzy和zbs的葡萄

Time limit: 1000ms Memory limit: 256Mb

Description

众所周知葡萄有红的，也有绿的。*zzy*喜欢吃红葡萄，*zbs*喜欢吃绿葡萄。

现在有 n 个葡萄排成一行，*zzy*和*zbs*想要分葡萄：把这一排 n 个葡萄分为 k 段，每段只有红葡萄或者绿葡萄，且 k 尽量的小。

然而，所有的葡萄都处在薛定谔态，他们想知道在随机意义下(每个葡萄有可能是红色或绿色，且概率相等)， k 的期望是多少。

Input Description

第一行一个整数 T 表示数据组数。 $(1 \leq T \leq 100)$

接下来 T 行每行一个整数 n 表示一排葡萄的数量。 $(1 \leq n \leq 10^6)$

Output Description

每行一个整数，对应期望 k 的整数部分。

Sample Input

```
3
1
2
3
```

Sample Output

```
1
1
2
```


1004 zbs和小布鞋的回文数

Time limit: 1000ms Memory limit: 256Mb

Description

某日，zbs心血来潮，给小布鞋出了一道关于小数的 $A - B$ 问题：

给定一个有 n 个元素的数组 A ,问有多少对 (i, j) 满足 $i < j$ ，且 $a_i - a_j$ 是一个整数。

小布鞋立马就给出了解法：把整数部分置0，直接记录每种小数出现的次数，再计算答案即可。

现在，zbs和小布鞋给你出了一道题目：

给定一个有 n 个元素的数组 A ，其中数组的元素 a_i 都是十进制正整数且都有 k 位，并且满足每个位上的数字都是1到4，问有多少对 (i, j) 满足 $i < j$ ，且 $a_i + a_j$ 是一个回文数。回文数即将数字从左到右写或者从右到左写都是一样的，如12341不是回文数，但12321是回文数。

Input Description

第一行一个整数 T ，代表数据组数。

接下来对于每组数据，第一行两个整数 n 和 k ，分别代表数组 A 的大小和每个元素的位数。

接下来一行有 n 个 k 位的整数代表数组 A 的元素，保证每位上的数字是1到4。

$1 \leq T \leq 10, 1 \leq n \leq 10^5, 1 \leq k \leq 12$ 。对于所有数据， n 的总和不超过 $5 * 10^5$ 。

Output Description

对于每组数据，输出一行，包括一个整数代表答案。

Sample Input

```
2
4 1
1 2 3 4
4 2
11 12 21 22
```

Sample Output

```
6
2
```

1005 zzy的数学水题

Time limit: 1000ms Memory limit: 256Mb

Description

zzy很喜欢数学题，所以他出了个大大大大大大水题来考考你。

大家一定见过这样子的函数， $f(m+n) = f(m) + f(n)$ 或 $f(mn) = f(m) * f(n)$ 。

这里我们给出一个函数，满足以下三个条件：

1. 对任意 $\gcd(m, n) = 1$, $f(mn) = f(m) + f(n)$ 。
2. 如果 x 只有一个质因子，那么 $f(x) = 1$ 。
3. $f(1) = 0$ 。

$\gcd(a, b)$ 表示 a 和 b 的最大公约数。

zzy想知道 $\sum_{i=1}^n f(i)$ 的值，聪明的你能告诉他吗。

Input Description

第一行一个整数 T 表示数据组数。 $(1 \leq T \leq 10)$

接下来 T 行每行一个整数 n 。 $(1 \leq n \leq 10^7)$

Output Description

对于每组数据，输出一个整数表示 $\sum_{i=1}^n f(i)$ 的值且独占一行。

Sample Input

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

Sample Output

```
0
1
2
3
4
```

1006 有趣的括号匹配

Time limit: 2000ms Memory limit: 256Mb

Description

括号匹配是一个很有趣的问题。我们称一串括号序列 s 是**完备的**，当且仅当括号序列 s 至少满足以下三个条件之一：

1. s 为空。
2. s 可被写成 (t) 的形式，其中括号序列 t 也是完备的。
3. s 可被写成 t_1t_2 的形式，其中括号序列 t_1, t_2 均是完备的。

以下括号序列均是完备的： $()$ 、 $((()))$ 、 $()()$ 、 $((()))()()()()$

以下括号序列均是不完备的： $)()$ 、 $)()()$ 、 $((()))()$

现提供如下两种操作：

- $shift(s)$ ：取出括号序列 s 的第一个括号，把它移到括号序列 s 的最后。
- $insert(s_1, s_2)$ ：把两个长度为 n 的括号序列 s_1, s_2 合并成一个长度为 $2n$ 的括号序列。其中，新序列的第 $2k - 1$ 项为 s_1 的第 k 项，新序列的第 $2k$ 项为 s_2 的第 k 项。

现给出两个长度为 n 的括号序列 s_1, s_2 。对于任意三元组 (a, b, c) ，对应的操作如下：

- 当 c 为0时，先进行 a 次 $shift(s_1)$ ，然后进行 b 次 $shift(s_2)$ ，最后执行 $insert(s_1, s_2)$
- 当 c 为1时，先进行 a 次 $shift(s_1)$ ，然后进行 b 次 $shift(s_2)$ ，最后执行 $insert(s_2, s_1)$

其中， a, b, c 均为非负整数且满足 $0 \leq a, b < n, c = 0$ 或 1 。试求：有多少对三元组 (a, b, c) 能使得最后生成的括号序列 s 是完备的。

Input Description

第一行一个整数 T ，表示数据组数。

对于每组数据，输入三行。第一行一个正整数 n ，表示括号序列的长度。

接下来两行分别为 s_1, s_2 ，代表两个括号序列。

$1 \leq T \leq 400, 1 \leq n \leq 1000$ 。对于所有数据，保证 n 的总和不超过 10^5 。

Output Description

对于每组数据，输出一行，包括一个整数表示答案。

Sample Input

```
3
2
((
))
2
()
()
4
()()
()()
```

Sample Output

4

2

8

1007 有趣的数组

Time limit: 1000ms Memory limit: 256Mb

Description

有两个长度为 n 的数组 A 和 B ，其中 B 数组的每个元素均为0。

现提供一种操作：选取 B 数组中任意一个长度大于1的区间，将这个区间中的每个数加上任意正整数 k 。

问：是否存在一种方案，通过若干次操作，把 B 数组变为 A 数组。如存在，输出"Yes"，否则输出"No"。

Input Description

第一行一个整数 T 表示数据组数。接下来每组数据占两行。 $(1 \leq T \leq 200)$

第一行一个整数 n 表示数组 a 的大小，接下来一行 n 个正整数，第 i 个数表示 a_i 的值。
($2 \leq n \leq 1000, 0 \leq a_i \leq 10^5$)

Output Description

对于每组数据，输出"Yes"或"No"，独占一行。

Sample Input

```
4
6
0 1 1 1 1 0
6
0 3 3 0 1 1
6
0 0 1 0 1 1
6
0 1 2 3 3 0
```

Sample Output

```
Yes
Yes
No
Yes
```

1008 火箭测试

Time limit: 2000ms Memory limit: 256Mb

Description

作为一个很厉害的火箭发明家，zzy每天都在测试场地亲自测试他设计的火箭。测试场地可以抽象为一个二维坐标系 xOy ，每次开始测试前，zzy都会站在原点 $(0, 0)$ 处。

zzy设计的火箭很有特点：作为一个功能单一的一次性用品，每枚火箭拥有两个属性值 a_i 和 b_i 。当zzy处于坐标 (x_i, y_i) 时使用这枚火箭，他会到达 $(x_i + a_i, y_i + b_i)$ 处。

然而zzy是个贪玩的人，他并不一定要测试完这 n 枚火箭的性能(毕竟用完就没了)，他只是想飞得尽量远而已。

作为zzy的得力助手(背锅侠)，你被要求解决如下问题：不妨设 (X, Y) 为zzy测试结束时所在位置，求离原点最远距离 $d = \sqrt{X^2 + Y^2}$ 的最大值。由于zzy不喜欢根号这种东西，你只需要输出 d^2 即可。

Input Description

第一行包含一个整数 $T(1 \leq T \leq 10)$ ，代表 T 次测试。

对于每次测试，第一行包括一个正整数 $n(1 \leq n \leq 1000)$ ，代表该次测试zzy携带了 n 枚火箭。

接下来 n 行，第 i 行包含两个用空格分开的整数 a_i 和 b_i ($-1000000 \leq a_i, b_i \leq 1000000$)，代表第 i 枚火箭拥有的两个属性值 a_i, b_i 。

Output Description

对于每次测试，输出一个正整数，代表你认为的答案。

Sample Input

```
2
10
1 1
4 5
1 4
1 9
1 9
8 1
0 -1
-2 -3
-4 -5
-6 -7
5
3 7
9 6
8 -5
-1 4
-2 -5
```

Sample Output

```
1097
505
```

1009 Au孙的奖牌

Time limit: 2000ms Memory limit: 256Mb

Description

众所周知，*au*孙获奖无数。

他的奖牌中包含金牌(用数字1代替)、银牌(用数字2代替)和铜牌(用数字3代替)。某天，*au*孙想要看看自己究竟获得了多少奖，于是他拿出了所有的奖牌，按一条直线整齐排列。这时他发现：若把这些奖牌看成是一个序列，那么该序列显然由若干个**尽量长**的“**仅包含同种奖牌**”的连续段构成。比如奖牌序列

1112233311，显然由 111、22、333、11 四个连续段构成。于是他向你提了一个问题：

在这些连续段中，最长的“仅包含金牌”的连续段有多长？

当然这个问题很naive，数一下就知道了。为了让问题变得有趣一点，*au*孙可以选择交换任意两个奖杯的位置(**仅限一次**)来使得答案尽量大。获奖无数的*au*孙显然知道这个问题的答案，他现在只是想考考你，你能回答他的这个问题吗？

Input Description

第一行一个正整数 T ，代表共 T 组测试用例。 $(1 \leq T \leq 10)$

对于每组测试用例，第一行包含一个正整数 n ，代表*au*孙拥有的奖牌个数。 $(1 \leq n \leq 100000)$

第二行包含一个长度为 n 的字符串，只包含数字1、2和3，代表*au*孙的奖杯排列情况。

Output Description

对于每个测试用例，输出一行，其中只包括一个正整数作为答案。

Sample Input

```
3
10
1121311121
10
1111111123
10
2112111131
```

Sample Output

```
5
8
7
```

1010 有趣的AB串

Time limit: 1000ms Memory limit: 256Mb

Description

众所周知，**回文串**就是正着读和反着读都一样的字符串，比如字符串`LEVEL`或者`NOON`就是回文串。而**子串**就是字符串中任意多个连续的字符组成的字符串。

现在，给定**有趣串** S 的定义：

1. S 串的长度大于1。
2. S 串均由字母 A 或 B 组成。
3. S 串中的每一个字符都蕴含在 S 串的某个长度大于1的回文子串内。

例如：`ABABBB`就是**有趣串**

第一个字符 A 蕴含在回文子串 `ABA` 中。

第二个字符 B 蕴含在回文子串 `ABA` 中。

第三个字符 A 蕴含在回文子串 `ABA` 中。

第四个字符 B 蕴含在回文子串 `BB` 中。

第五个字符 B 蕴含在回文子串 `BB` 中。

给定一个**有趣串** S ，请找出 S 的子串中有多少个**有趣串**。

Input Description

第一行一个整数 T ，表示表示数据组数。

接下来每组两行。第一行一个整数 N ，代表有趣串的长度。第二行为有趣串 S 。

$1 \leq T \leq 1000, 1 \leq N \leq 10^5$ 。对于所有的测试用例，保证 N 的总和不超过 10^6 。

Output Description

对于每组数据，输出一行，包含一个整数，代表有趣子串的个数。

Sample Input

```
3
5
ABABB
5
AAAAA
5
BBABB
```

Sample Output

```
6
10
6
```


1011 zzy爱下棋

Time limit: 1000ms Memory limit: 256Mb

Description

zzy很喜欢下棋，有一天他想出了这样一个问题：

给定一个 8×8 的棋盘，棋盘上的每个格子可能为空(记为0),可能有白子(记为1),可能有黑子(记为2)。

两个人轮流取子，每次轮到的人可以选定一个格子 (x, y) ，将 $\forall(i, j), x \leq i \leq 8, y \leq j \leq 8$ 的所有棋子全部拿走。

如果取走的棋子中**没有白子**,或者**有黑子**,即为负。通俗的说,为了保证该次操作不败,你至少取走一颗白子且不能取黑子。

两人轮流取子，问：先手是否有必胜策略。必胜输出"Yes"，否则输出"No"。

Input Description

第一行一个整数 T ，表示数据组数。 $(1 \leq T \leq 100)$

接下来每组数据8行，每行8个数字，0表示空，1表示白子，2表示黑子。

数据之间存在空行。

Output Description

对于每组数据，输出"Yes"或"No"，独占一行。

Sample Input

4
22222222
22222222
22222222
22222222
22222222
22222222
22222222
22222222

11111111
11111111
11111111
11111111
11111111
11111111
11111111
11111111

22222222
22222222
22222222
22222222
22222222
22222222

2 2 2 2 2 2 2 2

2 2 2 2 2 2 2 1

2 2 2 2 2 2 2 2

2 2 2 2 2 2 2 2

2 2 2 2 2 2 2 2

2 2 2 2 2 2 2 2

2 2 2 2 2 2 2 2

2 2 2 2 2 2 2 2

2 2 2 2 2 2 2 1

2 2 2 2 2 2 1 0

Sample Output

No

Yes

Yes

No

1012 有趣的比赛

Time limit: 1000ms Memory limit: 256Mb

Description

众所周知，参加 *ACM* 的队伍由三个人组成，但每支队伍只有一台电脑可用于写代码。不过，三个人可以通过协商解决这个问题。假若我们修改比赛规则，使得参加 *ACM* 比赛的每支队伍有 N 个人，每支队伍依然只能使用一台电脑，这样就会产生一个问题：同一时刻如果有多人想要写代码，他们就不得不协调一下谁先使用电脑了。

为了解决这个问题，zzy 拍脑瓜想出了这样一个办法：假设队伍有 N 个人，把队伍中的每个人标上编号 $1 - N$ 。如果同时有多个人想写代码，编号小的人优先使用电脑。

为了简化问题，我们假设每个人在整场比赛只做一道题，第 i 个人完成思考、可以写代码的时刻是 A_i ，第 i 个人写代码所需要的时间是 B_i 。在同一时刻，只能有一个人在使用电脑。忽略换人所花费的时间。

然而，这个方案并不公平，因为可能出现如下情况：编号为 N 的人是第二个完成思考的人，却有可能是最后一个写代码的人(因为编号小于 N 的人都准备好写代码，尽管他们完成思考的时间比较慢)。为了推翻 zzy 的方案，你需要计算每个人完成自己题目的**时刻**，假设初始时刻为 0。

Input Description

第一行一个整数 T ，表示数据组数。

每组数据占三行。第一行一个整数 N ，代表队伍的人数。

第二行 N 个整数。第 i 个整数表示编号为 i 的人准备好写代码的**时刻** A_i 。

第三行 N 个整数。第 i 个整数表示编号为 i 的人写完代码所需要的时间 B_i 。

$1 \leq T \leq 1000, 1 \leq N \leq 10^5, 1 \leq A_i \leq 10^9, 1 \leq B_i \leq 10000$

对于所有数据，保证 N 的总和不超过 200000。

Output Description

对于每组数据，输出一行，其中包括 N 个整数，第 i 个整数表示第 i 个人写完代码的时刻。注意行末**没有**空格。

Sample Input

```
3
3
1 5 2
4 1 1
3
3 2 1
2 2 2
3
0 0 0
1 2 3
```

Sample Output

5 6 7
5 7 3
1 3 6

1013 有趣的签到题

Time limit: 1000ms Memory limit: 256Mb

Description

这里有一道有趣的签到题。

给定整数 A, B, C, D 判断 $A^3 + B^3 + C^3$ 是否等于 D^3 。

如果等于输出 "Yes"，否则输出 "No"。

Input Description

第一行一个整数 T 表示数据组数。

接下来 T 行, 每行 4 个整数 A, B, C, D 。

$$1 \leq T \leq 1000$$

$$-1024 \leq A, B, C, D \leq 1024$$

Output Description

对于每组数据，输出 "Yes" 或 "No"。独占一行。

Sample Input

```
4
1 1 1 1
64 20 -20 64
1 6 8 9
7 8 9 10
```

Sample Output

```
No
Yes
Yes
No
```

Hint

Carefully!!!