

Problem A. 水质检测

Input file: 标准输入
Output file: 标准输出
Time limit: 1 秒
Memory limit: 128 MB

银江科技集团创建于 1992 年，作为城市大脑运营服务商，银江技术采用更智慧的方法，通过物联网、云计算、大数据、人工智能、区块链等核心自主技术的运用，致力于为智慧交通、智慧医疗、智慧建筑、智慧能源、智慧环境、智慧教育、智慧金融、智慧旅游等领域用户提供先进的产品。

例如，银江技术有着成熟的水质在线监测系统解决方案。现在你需要解决一个简化的情形，在一块小区域内有 n 个水质采样点，每个点会得到一个采样数据。现在给出一个衡量水质安全的阈值 k ，你需要对水质情况做出判断：

- 如果所有采样点采样数据均小于等于 k ，那么意味着这片水域水质良好；
- 如果存在采样点采样数据大于 k ，并且平均采样数据小于等于 k ，那么意味着这片水域合格；
- 如果存在采样点采样数据大于 k ，并且平均采样数据大于 k ，那么意味着这片水域水质不合格。

Input

第一行包含两个正整数 n, k ，其中 $1 \leq n \leq 50, 1 \leq k \leq 10^4$ 。

第二行包含 n 个整数，描述每个水质采样点得到的采样数据，第 i 个正整数 a_i ($1 \leq a_i \leq 10^4$) 描述第 i 个采样点的水质情况。

Output

如果水质良好，输出 “Good”；如果水质合格，输出 “OK”；如果水质不合格，输出 “Bad”。

Examples

标准输入	标准输出
3 3 1 2 3	Good

Problem B. MS 与美食街 2

Input file: 标准输入
Output file: 标准输出
Time limit: 1 秒
Memory limit: 128 MB

MS 是 Legendary Grandmaster DiDiDi 的神仙队友，有关他们如何在各种 GP 上捧杯的故事讲上三天三夜也讲不完。

有一天他们在 GP of Zhijiang 捧杯之后，MS 准备去枝江美食街研究一下今晚吃什么。

美食街可以看成是一个序列，MS 通过对美食街的顺序遍历，发现美食街一共有 n 个小吃店，每家店的小吃都各具特色。对于第 i 家小吃店，吃遍它家的小吃需要花费 $a^{c[i]}$ 元，即每家店小吃的费用都是同一个 a 的幂次。

注意到这个性质的 MS 开始思考，如果他想吃遍这个美食街，并且他携带的钱也得是 a 的幂次，那么他最少要携带多少钱。因为答案 a^k 可能很大，你只需要输出 k 即可。

Input

第一行包括两个整数 n, a ，保证 $1 \leq n \leq 50, 1 \leq a \leq 10^9$ 。

第二行包含 n 个整数，第 i 个整数表示 c_i ($0 \leq c_i \leq 10^9$)。

Output

输出一行包含一个整数 k ，表示最少需要携带 a^k 元。

Examples

标准输入	标准输出
3 10 5 6 3	7

Note

在上述样例中，三家店的小吃分别花费 10^5 元、 10^6 元、 10^3 元，三者之和为 1101000 元，故 MS 至少需要携带 10^7 元才能满足题目要求。

Problem C. MS 与矩阵填数

Input file: 标准输入
Output file: 标准输出
Time limit: 1 秒
Memory limit: 512 MB

对于一个 $n \times n$ 的矩阵 A ，定义 a_i 为矩阵第 i 行的最小值，即 $a_i = \min_{j=1}^n \{A_{i,j}\}$ ，这个矩阵的权值 $w(A)$ 为所有行最小值在 $1 \sim n$ 之间的行的最小值之和，即

$$w(A) = \sum_{i \in \{1,2,\dots,n\} \wedge a_i \in [1,n]} a_i.$$

现在给出一个正整数 n ，你可以通过在矩阵中不重复地填充 $1 \sim n^2$ 得到一个 $n \times n$ 的矩阵，对于所有可能得到的矩阵，你需要求出它们的权值之和。

Input

输入包含一个整数 n ，表示矩阵大小，数据保证 $n \leq 5000$ 。

Output

输出一行包含一个整数，表示所有矩阵的权值之和对 998244353 取模的值。

Examples

标准输入	标准输出
2	56

Note

2×2 的矩阵一共有 24 种，其中本质不同的情况只有以下两种：

情况 1: $\begin{matrix} 1 & 2 \\ 3 & 4 \end{matrix}$ 情况 2: $\begin{matrix} 1 & 3 \\ 2 & 4 \end{matrix}$

其中，第一种矩阵权值为 1，共有 8 个，第二种矩阵价值为 3，共有 16 个，因此总权值为 $8 \times 1 + 16 \times 3 = 56$ 。

Problem D. MS 与矩阵取数

Input file: 标准输入

Output file: 标准输出

Time limit: 1 秒

Memory limit: 128 MB

MS 是个图论爱好者，不过他只会套板子。他做了方格取数这道题后觉得很有意思，他也想出一道有趣的题，比如有没有可能在 AC 自动机上跑最大流之类的。

为了出题，MS 找来一个 3×3 的方格，给每个格子填充上 0 和 1，他想起一个以前玩过的游戏。

每次选择一个方格，让他和相邻的四个格子（也就是一个“十”字形）翻转，问怎么操作能让整个方格的数字相同，比如他选择了中间的格子，翻转情况就如下图：

0	1	0		0	0	0
0	1	1	→	1	0	0
1	0	1		1	1	1

他玩了几局觉得太简单了，于是他决定给每个位置填充上一个 $[0, 9]$ 的整数，每次操作让一个格子和相邻的四个格子上的数字加上 1，特别地 9 加上 1 会变成 0，要使得方格中的所有数相同。

5	1	4		5	2	4
1	9	1	→	2	0	2
9	8	1		9	9	1

虽然还是很简单，但 MS 想让你也试试，显然选择格子的顺序是没有影响的，所以请你输出每个格子被选择的次数。

Input

第一行包含一个整数 T 表示数据组数，保证 $T \leq 100000$ 。

接下来 T 行，每行表示一组数据，包含 9 个整数表示一个 3×3 的方格。

Output

每组数据输出一行，如果有解，输出 9 个整数 (≤ 9) 表示每次选择的格子，否则输出 -1 。

注意解可能不唯一，本题提供 Special Judge，你只需要输出任意一组合法解即可。

Examples

标准输入	标准输出
1 5 1 4 1 9 1 9 8 1	1 9 3 1 2 0 1 5 0

Note

给出的输入样例即为题目描述中的第二张插图中的例子。

Problem E. 招募军队

Input file: 标准输入

Output file: 标准输出

Time limit: 1 秒

Memory limit: 128 MB

为了抵御其他国家的侵略，P 国决定对自己所统治的 n 座城市招募军队。由于每个城市的经济水平不一样，所以不同城市有着自己的招募人员的费用。

对于城市 i ，其直接招募军队的费用是 a_i 。如果城市 i 与城市 j 建交，城市 i 还可以通过私下关系从城市 j 间接招募军队，费用为 $w_{i,j}$ ，当然前提是城市 j 已经直接或间接招募了军队。

你需要最大化实际费用 c 与 $\sum_{i=1}^n a_i$ 差值的绝对值，约定每个城市只需招募一次军队。

Input

第一行包含两个整数 n, m ($1 \leq n \leq 10^5, 1 \leq m \leq \min(10^5, \frac{n(n-1)}{2})$)，表示有 n 座城市， m 个建交关系。

第二行包含 n 个用空格隔开的整数， a_i 表示从城市 i 招募军队的费用，保证 $1 \leq a_i \leq 10^9$ 。

接下来 m 行，每行包含 3 个整数 $i, j, w_{i,j}$ ，表示城市 i 与城市 j 建交并且城市 i (或者城市 j) 互相从对方城市招募军队的费用为 $w_{i,j}$ ，保证 $1 \leq w_{i,j} \leq 10^9$ 。

Output

输出共一行，包含一个整数，表示所求的最大差值。

Examples

标准输入	标准输出
3 2 2 3 5 1 2 1 2 3 10	7
6 3 1 3 3 6 6 10 5 6 1 1 2 1 2 3 1	13

Note

对于第一个样例，总共有 3 个城市，招募费用分别为 2,3,5，具体解释如下：

- 实际费用最小为 8：城市 1 和 3 在自己的地盘招募军队，城市 2 通过私下关系从城市 1 招募军队；
- 实际费用最大为 17：城市 1 和 3 在自己的地盘招募军队，城市 2 通过私下关系从城市 3 招募军队；
- 因此与总和 $2 + 3 + 5 = 10$ 的最大差值为 7。

Problem F. DiDiDi 与孪生素数

Input file: 标准输入
Output file: 标准输出
Time limit: 1 秒
Memory limit: 128 MB

DiDiDi 是一个喜欢素数的女孩子。

MS 知道她的小爱好，于是送了一堆孪生素数给她。

DiDiDi 很快就算出了第十万对孪生素数是 (18409199, 18409201)，现在她想知道第 n 对孪生素数的和除以 12 的余数是多少。

如果 p 与 q 是一对孪生素数，那么 p, q 都是素数，且 $q - p = 2$ 。

Input

输入共一行，包含一个整数 n ，保证 n 在 32 位有符号整数范围内，即 $1 \leq n < 2^{31}$ 。

Output

一个数字，表示第 n 对孪生素数除以 12 的余数。

Examples

标准输入	标准输出
2	0
3	0

Note

第二对孪生素数是 (5, 7)， $(5 + 7) \bmod 12 = 0$ 。

第三对孪生素数是 (11, 13)， $(11 + 13) \bmod 12 = 0$ 。

Problem G. 圆圆 circle

Input file: 标准输入

Output file: 标准输出

Time limit: 2 秒

Memory limit: 128 MB

圆环上任意两点间的路径都永远不唯一，在达成目的之前，我们面临着选择。一切美好的事物都是曲折地接近自己的目标，一切笔直都是骗人的，所有真理都是弯曲的，时间本身就是一个圆圈。尼采的“永恒回归”理论析构了选择的价值，而假如以此标准审视，在另一个平行时空，liella 也有可能更名为圆圆 circle。

Oewt 开始探寻选择的意义。

具体而言，他构造了一个立有 n 根木柱的圆环，木棒的编号从 $1 \sim n$ ，并按照顺序构成一个圆环。他可以选择顺时针从一根起始棒 a_i 开始在每根木柱上放置小圆环，直至终止棒 b_i 为止（ b_i 不放），或者逆时针做这个操作（ a_i 不放）。

由于操作数比较多，Owet 把操作数压缩后以一个三元组 (a_i, b_i, c_i) 给出，表示有 c_i 个操作，要从 a_i 选择顺时针或者逆时针放到 b_i 。

Oewt 想知道在做完这些操作之后，圆环上所有木柱中套的小圆环最多的那根套的小圆环数最小为多少。

Input

第一行包含两个正整数 n, m ($n \leq 2 \times 10^5, m \leq 10^5$)，表示木棒数和选择次数。

接下来 m 行，每行包含三个非负整数 a_i, b_i, c_i ，表示起始棒，终止棒和操作次数，保证 $1 \leq a_i, b_i \leq n$ ， $0 \leq c_i \leq 10^9$ 。

Output

输出一行包含一个整数，表示圆环上所有木柱中套的小圆环最多的那根套的小圆环数的最小值。

Examples

标准输入	标准输出
4 2 1 3 3 1 3 1	2

Note

第一个操作一个逆时针放，两个顺时针放，第二个操作逆时针放。

顺时针放，1，2，3 被放了两次，逆时针放，4 被放了两次，故放小圆环最多的被放了两个。

Problem H. 基础离散数学练习题

Input file: 标准输入

Output file: 标准输出

Time limit: 1 秒

Memory limit: 128 MB

在离散数学中，我们定义命题是可以唯一确定真值的语句，即要么为真，要么为假。

现在，我们有 n 个形如以下形式的命题：

- 在这 n 个命题中，恰好有 x_1 个命题为真。
- 在这 n 个命题中，恰好有 x_2 个命题为真。
- 在这 n 个命题中，恰好有 x_3 个命题为真。
-
- 在这 n 个命题中，恰好有 x_n 个命题为真。

现在，rebmit 想知道，这 n 个命题中，最多有几个命题可以同时成立。如果存在矛盾，无法找到这样的答案，则输出 -1 。

Input

第一行包含一个正整数 n ($1 \leq n \leq 10^5$)，描述命题的个数。

第二行包含 n 个非负整数，第 i 个非负整数 x_i 描述第 i 个命题，保证 $0 \leq x_i \leq 10^5$ 。

Output

如果有解，输出一个非负整数表示答案，如果无解则输出 -1 。

Examples

标准输入	标准输出
4 0 1 2 3	1
1 0	-1

Note

对于第一组样例，第二个命题为真，其他都为假，不难说明不存在更大的答案满足题意。

对于第二组样例，不论该命题为真或假，都自相矛盾，故为 -1 。

Problem I. 翻牌

Input file: 标准输入

Output file: 标准输出

Time limit: 1 秒

Memory limit: 128 MB

为了在游戏领域一较长短，Steve 和 Hugin 决定玩翻牌游戏。

他们将 n 张相同的牌按序排成一列，每张牌有正反两面，初始时所有牌都是正面朝上。Steve 和 Hugin 两个人会轮流进行翻牌操作，每次翻牌可以选择一个长度为 k 的连续区间，然后翻转这个区间里的所有牌（也就是说将正面朝上的牌翻转为反面朝上，反之亦然），需要注意的是这个区间最左侧的牌必须是正面朝上的。如果一个人没办法再进行翻牌，那么他就输了。

已知 Steve 先手，请问在两个人都采取最优策略的情况下，对于给定的 n, k ，谁一定能够获胜。

Input

第一行输入一个正整数 T ($1 \leq T \leq 3 \times 10^5$) 代表数据组数。

输入 T 行，每行两个正整数 n, k ($1 \leq n, k \leq 10^9$)，意义如题面所述。

Output

对于每组测试数据输出一行，如果 Steve 必胜，请输出 **Steve**；如果 Hugin 必胜，请输出 **Hugin**。

Examples

标准输入	标准输出
2	Hugin
4 2	Steve
2 2	

Note

对于 $n = 4, k = 2$ 这个样例，Steve 必败；总共三种情况：

- Steve 先手选择翻转区间 $[1, 2]$ ，那么 Hugin 只需要翻转 $[3, 4]$ 就可以结束游戏；
- Steve 先手选择翻转区间 $[3, 4]$ ，那么 Hugin 只需要翻转 $[1, 2]$ 就可以结束游戏；
- Steve 先手选择翻转区间 $[2, 3]$ ，Hugin 翻转 $[1, 2]$ ，Steve 只能翻转 $[2, 3]$ ，Hugin 翻转 $[3, 4]$ 结束游戏。

对于 $n = 2, k = 2$ 这个样例，Steve 必胜；Steve 可以先手翻转区间 $[1, 2]$ 就可以结束游戏。

Problem J. Lecxcy 与位运算

Input file: 标准输入
Output file: 标准输出
Time limit: 1 秒
Memory limit: 128 MB

教练说：“要造题。”就有了题。

Lecxcy 又不知道从哪里整来了一个长度为 n 的序列 a 。现在他想知道 $\sum_{1 \leq i < j \leq n} a_i \oplus a_j$ 的值，其中 \oplus 表示按位异或运算。

由于答案可能很大，你只需要输出答案模 998244353 的值。

Input

第一行包含一个正整数 n ($2 \leq n \leq 10^6$)，表示序列 a 的长度。

第二行 n 个用空格隔开的整数，表示序列 a 中的元素，数据保证 $0 \leq a_i < 2^{30}$ 。

Output

输出共一行，包含一个整数表示答案。

Examples

标准输入	标准输出
5 1 2 3 4 5	42

Note

关于按位异或运算：两个数作按位异或操作，就是对每一位单独进行考虑，如果两个数对应二进制位相同，则所得结果对应位的值为 0，如果两个数对应二进制位不同，则所得结果对应位的值为 1。

关于样例解释：

$$1 \oplus 2 = 3, 1 \oplus 3 = 2, 1 \oplus 4 = 5, 1 \oplus 5 = 4$$

$$2 \oplus 3 = 1, 2 \oplus 4 = 6, 2 \oplus 5 = 7$$

$$3 \oplus 4 = 7, 3 \oplus 5 = 6$$

$$4 \oplus 5 = 1$$

于是可知总和为 42。

Problem K. KoiKoi~

Input file: 标准输入

Output file: 标准输出

Time limit: 1 秒

Memory limit: 128 MB

包桑在看了《夏日大作战》之后对花札产生了兴趣。

通过百度，她知道了一副花札一共有 48 张，每 4 张构成一个月，一共 12 个月，分别为：

睦月：松间鹤（光牌）、松间赤短（短牌）、松（皮牌）*2

如月：梅间莺（种牌）、梅间赤短（短牌）、梅（皮牌）*2

弥生：樱上幕帘（光牌）、樱间赤短（短牌）、樱（皮牌）*2

卯月：藤上鹈（种牌）、藤上短册（短牌）、藤（皮牌）*2

皋月：蒲间八桥（种牌）、蒲上短册（短牌）、菖蒲（皮牌）*2

水无月：牡丹蝴蝶（种牌）、牡丹青短（短牌）、牡丹（皮牌）*2

文月：萩间野猪（种牌）、萩上短册（短牌）、萩（皮牌）*2

叶月：芒上月（光牌）、芒上雁（种牌）、芒（皮牌）*2

长月：菊与酒（种牌）、菊上青短（短牌）、菊（皮牌）*2

神无月：枫间鹿（种牌）、枫上青短（短牌）、枫（皮牌）*2

霜月：柳间小野道风（光牌）、柳上燕（种牌）、柳上短册（短牌）、柳雷雨鼓（皮牌）

师走：桐上凤凰（光牌）、桐*3（皮牌）



每局每人按规则摸牌，而当手牌满足以下任意役种时，即可宣布结束游戏。（没有特殊说明，点数可叠加）

五光：拥有五张光牌，记10点

四光：拥有除了柳间小野道风之外的四张光牌，记8点

雨四光：拥有包含柳间小野道风的四张光牌，记7点

三光：拥有三张光牌，记5点

上面四种役种不能叠加，结算时取最大。

饮：拥有芒上月、樱上幕帘、菊与酒，记5点

花见酒：拥有樱上幕帘、菊与酒，记3点

月见酒：拥有芒上月、菊与酒，记3点

以上三种役种不能叠加，结算时取最大。

松桐坊主：拥有松间鹤、芒上月、桐上凤凰，记5点

猪鹿蝶：拥有荻间野猪、枫间鹿、牡丹蝴蝶，记5点

表菅原：拥有松间鹤、梅间莺、樱上幕帘，记5点

草：拥有藤上短册、蒲上短册、荻上短册(短牌)，记3点

赤短：拥有松间赤短、樱间赤短、梅间赤短，记5点

青短：拥有牡丹青短、菊上青短、枫上青短，记5点

月牌：拥有同一个月的四张牌，记4点，拥有多个同月只计算一次

短：拥有五张短牌，记1点，之后每多一张多记1点

种：拥有五张种牌，记1点，之后每多一张多记1点

皮：拥有十张皮牌，记1点，之后每多一张多记1点

五鸟：拥有梅间莺、藤上鹑、芒上雁，记5点

现在包桑手上已经摸了一副花牌，请帮她计算这副花牌一共多少点。

Input

第一行包含一个正整数 n ($n \leq 24$)，表示一共有多少张花牌。

第二行给出 n 张花牌的牌面，用 mm-kth 表示，严格按照上面给出的顺序，比如 1-1 是松间鹤，1-2 是松间赤短，1-3、1-4 是松，12-1 是桐上凤凰。不保证输入的牌按序递增。

Output

输出一行，包含一个整数，表示总点数。

Examples

标准输入	标准输出
5 1-1 3-1 8-1 11-1 12-1	15

Problem L. 骰子 2

Input file: 标准输入

Output file: 标准输出

Time limit: 2 秒

Memory limit: 128 MB

Sleep_Zzz 在知乎上看到了一个简单面试题，但是他对这个题目解法非常不满意，于是他和奥尔兹·hxy 大佐开始考虑优化这个问题。

我今天看到一个面试题，问 n 个筛子点数之和等于 m 的概率。简单的做法是 $n*m$ 的 dp。这个题是不是可以用 fft+倍增做到 $\log(n)*\log(m)*m$?

说不定能更低捏

不过很快他们发现自己傻逼了，这个题目随便推导一下就可以做到 $O(n)$ 复杂度，于是他们拿着骰子开始玩博弈游戏。

Sleep_Zzz 拿了 n 个六面骰，而 DFHXY 则拿了 $\frac{n}{2}$ 个十二面骰，两个人开始比谁掷出来的总点数大。

DFHXY 觉得这个游戏并不公平，因为两个六面骰至少能骰出两点，但是一个十二面骰可能骰出一点。

经过一番友好协商，Sleep_Zzz 答应了如果两人骰出总点数一致也算 DFHXY 胜利，而你则需要帮 DFHXY 计算他的胜率。

Input

输入共一行，包含一个整数 n 表示 Sleep_Zzz 拿了 n 个骰子 ($n \leq 5 \times 10^4$)，保证 n 是偶数。

Output

输出共一行，包含一个整数表示 DFHXY 胜利的概率对 998244353 取模后的值。

Examples

标准输入	标准输出
2	499122177

Note

两个六面骰可以掷出 1 ~ 12 点的概率分别为 $0, \frac{1}{36}, \frac{2}{36}, \frac{3}{36}, \frac{4}{36}, \frac{5}{36}, \frac{6}{36}, \frac{5}{36}, \frac{4}{36}, \frac{3}{36}, \frac{2}{36}, \frac{1}{36}$ 。

一个十二面骰可以掷出 1 ~ 12 点的概率分别为 $\frac{1}{12}, \frac{1}{12}, \frac{1}{12}, \frac{1}{12}, \frac{1}{12}, \frac{1}{12}, \frac{1}{12}, \frac{1}{12}, \frac{1}{12}, \frac{1}{12}, \frac{1}{12}, \frac{1}{12}$ 。

胜利的概率为 $\frac{1}{12} \cdot \frac{1}{36} + \frac{1}{12} \cdot (\frac{1}{36} + \frac{2}{36}) + \cdots + \frac{1}{36} \cdot 1 = \frac{1}{2}$ ，对 998244353 取模即为 499122177。

Problem M. 智慧交通

Input file: 标准输入
 Output file: 标准输出
 Time limit: 4 秒
 Memory limit: 512 MB

银江科技集团有限公司作为浙江省最大的智能化系统集成商，是中国城市智能交通行业的领军企业之一。

众所周知，ZJUT 非常狭长，所以引入智慧交通系统势在必行，为了简化问题，我们假设校园是一条自西向东分布着 n 座建筑物的链，其中每座建筑物都有一个同学（因为疫情，一栋楼住且仅住一个人），建筑间有 m 条道路，第 i 条连接着第 l_i 栋建筑和第 r_i 栋建筑。由于计算机教学楼分布在生活区的东部，所以计算机系的同学们上课只会从西边往东边走。

已知这一天有 d 节课，第 i 节在建筑 x_i 上课，在 x_i 西边（寝室里窝着）的同学都要赶去上课，并且 k_i 名同学（分别居住在第 $x_{i,1}, x_{i,2}, \dots, x_{i,k_i}$ 栋建筑）不需要上这门课，由于这天下雨了，同学们都倾向于走经过建筑物最多的路径，很显然这导致了拥堵。（每节课独立考虑，不用考虑 x_i 东侧的学生）。

智慧交通系统的策划人现在想知道每节课经过建筑物最多的同学经过了多少栋建筑物，这项指标对于衡量最长等待时间很有意义。

作为一位成熟的 ACM 选手，你需要计算出这些数据，并将它交给策划人。

Input

第一行包括三个整数 n, m, d ，表示 n ($1 \leq n \leq 10^5$) 座建筑物， m ($0 \leq m \leq 2 \times 10^5$) 条道路，一天有 d ($1 \leq d \leq 10^5$) 节课。

接下来 m 行，每行有两个整数 l_i, r_i ，表示这条道路连接着第 l_i, r_i 栋建筑，保证 $1 \leq l_i, r_i \leq n$ 。

接下来 d 行，每行首先包括整数 x_i 和 k_i 分别表示第 i 节课在建筑 x_i 上，这节课有 k_i 个同学不需要上课，之后 k_i 个数表示不需要上课的同学分别住在哪个寝室，数据保证 $0 \leq k_i \leq n$ 并且 $\sum_{i=1}^d k_i \leq 10^5$ 。

Output

输出共有 d 行，每行一个整数，表示每节课要经过建筑物最多的同学最多经过了几栋建筑物，如果没有同学要上课，请输出 No Student!。

Examples

标准输入	标准输出
5 4 3	2
1 2	1
2 3	No Student!
3 4	
4 5	
4 2 1 2	
3 2 1 2	
3 3 1 2 3	

Note

在样例中，第一节课在建筑 4 上课，1,2 都不来上课，只有 3,4 上课，其中 3 上课要经过 3,4 两栋建筑，所以答案是 2。

第二节课在建筑 3 上课，1,2 都不来上课，只有 3 上课，3 上课经过 3 一栋建筑，所以答案是 1。

第三节课在建筑 3 上课，1,2,3 都不来上课，所以输出 No Student!。