

第三届环球杯



第 7 赛段华沙

2024 年 8 月 24-25 日

该问题集应包含 13 个问题（A 至 M），共 17 页。



根据

Akademickie Mistrzostwa Polski w Programowaniu Zespoowym (AMPPZ)

主办方





问题 A. 公交分析

时间限制 2 秒 内存限制
1024 兆字节

亚当要乘坐公共汽车开始一段很长的旅程。他可以选择购买 2 波兰兹罗提的 20 分钟车票和 6 波兰兹罗提的 75 分钟车票¹。车票可在任意整点购买，并自动生效。例如，9:14:00 购买的 20 分钟车票有效期至 9:33:59。即使亚当仍持有有效车票，他也可以购买新车票。

公交车在 N 个站点停靠，第 i 个站点在第 t_i 分钟停靠 _{i} 。在每个站点，检票人员可能正在等待。他们上车后立即检查所有乘客，然后在同一站点下车（这一过程只需几秒钟）。

在旅途的一开始，亚当就会收到哪些车站有检票员在等候的信息。然后，他将计划购票，以尽量减少总支出，并在每次检票时持有有效车票。

考虑检票员安置的所有 2^N 方案，计算亚当的最优支出（波兰兹罗提）。输出除以 $10^9 + 7$ 。

输入

输入的第一行包含一个整数 N ($1 \leq N \leq 1000$) -- 停顿数。

第二行包含 N 个整数 t_1, t_2, \dots 的递增序列。 , t_N ($1 \leq t_i \leq 10^9, t_i < t_{i+1}$) -- 公交车停靠的分钟数。

输出

输出一个整数 - 结果的余数除以 $10^9 + 7$ 。

实例

标准输入	标准输出
3 1 8 20	14
5 25 45 65 85 1000000000	156

备注

在第一个测试案例中，有 3 个站点和 8 种情况需要考虑。如果亚当收到的信息是任何站点都没有检票员，那么他就不会购买任何车票。在其余 7 种情况下，他只需要一张 20 分钟的车票，票价为 2 波兰兹罗提。答案是 $0 + 7 \cdot 2 = 14$ 波兰兹罗提。

让我们来看看第二个示例测试用例中 32 个场景中的 2 个：



- 所有 5 个车站（25、45、65、85、1 000 000 000）的检票员 - 亚当花费 8 波兰兹罗提：6 波兰兹罗提购买一张 75 分钟的车票，2 波兰兹罗提购买一张 20 分钟的车票，例如在 25 分钟和 1 000 000 000 分钟购买的车票。
- 2 个车站（25、45）的检票员 - Adam 花 4 波兰兹罗提购买两张 20 分钟的车票，例如在 25 和 45 分钟或 17 和 26 分钟购买的车票。

¹华沙的实际票价为 20 分钟 3.40 波兰兹罗提，75 分钟或全程 4.40 波兰兹罗提。通常情况下，购买更长时间的车票是最佳选择。



问题 B. 边界缺失

时间限制 4 秒 内存限制
1024 兆字节

一个整数区间 $[1, L]$ 被分割成 N 个非空整数区间 $[a_i, b_i]$ ($1 \leq a_i \leq b_i \leq L$, for $1 \leq i \leq N$)，使得 $[1, L]$ 中的每个整数都恰好属于区间 $[a_i, b_i]$ 中的一个。然后，得到的区间 $[a_i, b_i]$ 的部分边界被隐藏起来。

给你一组区间，其中有些边界可能缺失。您的任务是确定这些区间是否可以通过上述方法产生，即是否有可能通过替换缺失值的方式使这些区间非空、成对不相交，并且共同覆盖从 1 到 L 的所有整数。

输入

输入的第一行包含一个整数 T ($1 \leq T \leq 30\,000$)，表示测试用例的数量。然后是 T 个测试用例的描述，一个接一个。

每个测试用例描述的第一行包含两个整数 N 和 L ($1 \leq N \leq 200\,000$, $1 \leq L \leq 10^9$)，分别表示给定区间的数量和将被分割成若干部分的原始区间的长度。下面 N 行中的每一行都包含两个整数， a_i 和 b_i ($-1 \leq a_i, b_i \leq L$; $a_i, b_i \neq 0$)。数字 -1 表示缺失值。如果 a_i 和 b_i 都是正数，则 $a_i \leq b_i$ 。

所有测试用例中数字 N 的总和不大于 200 000。

输出

输出结果应正好包含 T 行。第 i 行应包含第 i 个测试用例的答案：如果输入中描述的区间可以用任务中描述的方法生成，则答案为 TAK；如果不可能，则答案为 NIE。

示例

标准输入	标准输出
3	TAK
4 51	NIE
1 -1	NIE
11 50	
-1 -1	
-1 10	
3 2	
-1 -1	
-1 -1	
-1 -1	
2 3	
1 2	



备注

在第一个测试用例中，我们有 $L = 51$ ，区间 $[1, 51]$ 可以分割成区间 $[1, 7]$ 、 $[8, 10]$ 、 $[11, 50]$ 和 $[51, 51]$ 。隐藏一些值后，这些区间与给定的区间 $[1, -1]$ 、 $[-1, 10]$ 、 $[11, 50]$ 和 $[-1, -1]$ 相匹配。因此，本测试用例的答案是 TAK。

在第二个测试案例中，我们有 3 个区间，所有区间的边界都是隐藏的。然而，区间 $[1, 2]$ 只包含两个整数，因此不可能将其分割成 3 个非空且不相交的整数区间。因此，这个测试案例的答案是 NIE

。



在第三个测试案例中，我们有两个区间 $[1, 2]$ 和 $[2, 3]$ ，并且没有缺失边界。数字 2 被覆盖了两次，因此答案是 NIE。



问题 c. 价格组合

时间限制

2 秒 内存限制

1024 兆字节

Czesaw 决定自己制作一台电脑。经过深思熟虑，他选择了一套完整的配置，并列出了需要购买的 N 个组件清单。

该地区有两家电子产品商店：AutoAGD 和 BioBezpieczniki。Czesaw 清单上的第 i 个元件在 AutoAGD 商店的价格为 A_i 波兰兹罗提，在 BioBezpieczniki 商店的价格为 B_i 波兰兹罗提。每家商店都独立提供 "价格组合：买一送一" 促销活动--购买两件产品时，便宜的那件免费（以任何方式结算）。促销不受限制，这意味着同一顾客可在一家商店购买任意多件商品，每次都可享受促销。

帮助 Czesaw 以尽可能低的总成本购买所有部件。输出尽可能少的 费用总和。

输入

输入的第一行包含一个整数 N ($1 \leq N \leq 200\,000$)，表示 Czesaw 需要的组件数量。

第二行包含 N 个整数 A_1, A_2, \dots, A_N ($1 \leq A_i \leq 10^9$)，表示 AutoAGD 商店中组件的价格。

同样，第三行包含 N 个整数 B_1, B_2, \dots, B_N ($1 \leq B_i \leq 10^9$)，表示 BioBezpieczniki 商店的价格。

输出

输出一个整数 - Czesaw 在购买所有 N 组件

示例

标准输入	标准输出
7 10 12 19 99 10 8 49 9 14 15 199 11 7 19	131

备注

Czesaw 想购买 7 个部件。他可以购买以下物品：

- 在 AutoAGD 中，他购买组件 1（10 兹罗提）和组件 5（10 兹罗提），支付 10 兹罗提。
- 在 AutoAGD 中，他购买了部件 4（99 兹罗提）和部件 7（49 兹罗提），支付了 99 兹罗提。
- 在 BioBezpieczniki，他购买了 2 号部件（14 兹罗提）和 3 号部件（15 兹罗提），支付了 15 兹



罗提。

- 在 BioBezpieczniki，他以 7 波兰兹罗提的价格购买了 6 号元件。

所有消费共计 $10 + 99 + 15 + 7 = 131$ 波兰兹罗提。



问题 D. 数据确定

时间限制 2 秒 内存限制
1024 兆字节

曾经有一位非常严谨的科学家，决定一劳永逸地回答关于生命、宇宙和万物的终极问题。他从理论上开始思考，最终得出结论：答案是一个正整数，等于 m 。然而，这些思考是建立在许多关于生命、宇宙和万物的不确定假设之上的。理论上的考虑应该得到实验证据的支持！

科学家设计了一个特殊的实验，其中存在各种测量误差。他进行了 n 次实验，第 i 次实验的结果是数字 a_i 。在他的科研工作中，他计划包含恰好 k 次实验的数据，而这些数据的中位数²必须恰好是 m ，才能证实他的理论。

验证他能否实现目标。编写一个程序，在给出所有 n 个实验结果的情况下，判断是否有可能从中选择 k 个实验，使其结果的中位数正好是 m 。

输入

输入的第一行包含一个整数 t ($1 \leq t \leq 10\,000$)，表示要考虑的独立情景的数量。每种情况分两行描述。

在每个方案的第一行，有三个整数 n 、 k 和 m ($1 \leq k \leq n \leq 200\,000$, $1 \leq m \leq 10^9$)，分别代表已进行的实验数、科学工作所需的实验数和期望的中位数。第二行包含 n 个整数 a_1, \dots, a_n ($1 \leq a_i \leq 10^9$)，表示实验结果。

所有测试用例中的数值 n 之和不超过 200 000。

输出

输出结果应由 t 行组成，包含每个方案的答案。在第 i 行中，如果可以在第 i 个方案中选择适当的 k 个实验，则应有一个单词 "TAK"，否则应有一个单词 "NIE"。

示例

标准输入	标准输出
3	TAK
6 4 42	NIE
41 43 41 57 41 42	NIE
4 2 4	
1 2 5 8	
7 5 57	
101 2 42 5 57 7 13	

备注



在第一种情况下，你可以选择结果为 (41, 43, 41, 57) 的实验；排序后得到序列 (41, 41, 43, 57)，其中中间两个元素的算术平均数为 $\frac{43+41}{2} = 42$ 。

在第二种情况下，不可能选择中位数为 4 的一对元素：

- 序列 (2, 5) 的中位数为 $\frac{2+5}{2} = 3.5$ ，过低。
- 另一方面，序列 (1, 8) 的中位数为 $\frac{1+8}{2} = 4.5$ ，过高。

²序列的 *中位数* 是排序后的中间元素。如果序列的长度是偶数，中位数就是中间两个元素的算术平均数。例如，序列 (9, 7, 3, 4, 5) 的中位数是 5，序列 (3, 1, 6, 6) 的中位数是 $\frac{3+6}{2} = 4.5$ 。



问题 E. 快速驱逐

时间限制 4 秒 内存限制
1024 兆字节

Bajtocja 是一块长方形土地，由 $H \times W$ 单元组成， H 行 W 列。每个单元的边界都是一条交通稀少的地方道路。每个单元要么空着，要么只有一个居民居住。居民们喜欢安静，不愿意住在规划中的快速路旁边³。

您的任务是规划从 Bajtocja 左上角到右下角的快速路路线。该路线只能沿现有道路行驶，以尽量减少重建成本。转弯的长度和数量并不重要。

要驱逐每一个在快速路边上，甚至是角落里的居民。至少需要驱逐多少居民呢？

输入

输入的第一行包含两个数字 H 和 W ($1 \leq H, W \leq 50$) --Bajtocja 的尺寸。

接下来的 H 行描述了有人居住和无人居住的单元格。每行包含一个长度为 W 的字符串，由字符"." (空单元格) 和 "#" (有人居住的单元格) 组成。

输出

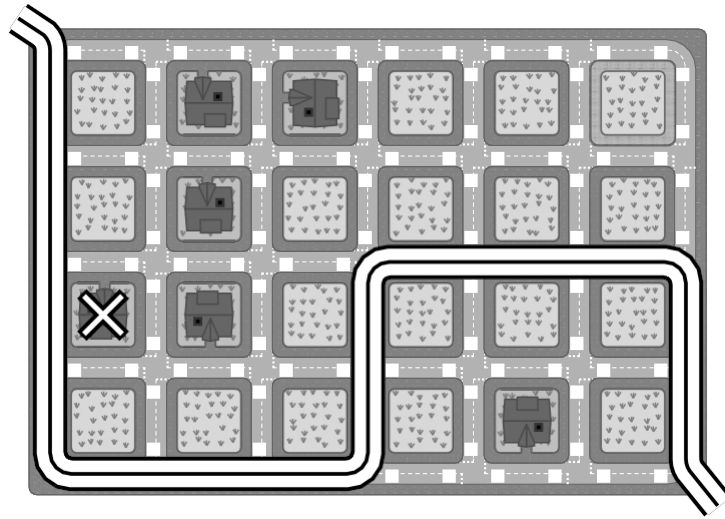
输出一个整数，即被驱逐居民的最少人数。

示例

标准输入	标准输出
4 6 ...### ...#... ...##... ...#..	1

备注

如果将居民从第一列和第三行的单元格中驱逐出去，就可以建造出如下的快速路：



³很多年前，一位老人非常恼火，他在自己的房子上绑了很多气球，然后飞走了。



问题 F. 斐波那契融合

时间限制 4 秒 内存限制
1024 兆字节

斐波纳契数是 0、1、1、2、3、5、8、13、21、.....。 - 每个数字（前两个数字除外）都是前两个数字之和。
给你 N 个数 a_i 。数出 $i < j$ 的数对，使得 $a_i + a_j$ 是斐波纳契数。

输入

输入的第一行包含一个整数 N ($1 \leq N \leq 200\,000$)。
接下来的 N 行中，每一行都包含一个整数 a_i ($1 \leq a_i < 10^{2000\,000}$)。这些数字的总位数不超过 5 000 000。

输出

输出一个整数 - 与斐波那契数字相加的数对的个数。

示例

标准输入	标准输出
6 50 8 8 5 72 354224848179261915070	4

备注

这样的对子有 4 对：

- $a_2 + a_4 = 8 + 5 = 13$
- $a_3 + a_4 = 8 + 5 = 13$
- $a_1 + a_4 = 50 + 5 = 55$
- $a_4 + a_6 = 5 + 354224848179261915070 = 354224848179261915075$



问题 G. 天才游戏

时间限制 2 秒 内存限制
1024 兆字节

两个天才正在玩下面的游戏。他们有一个 $n \times n$ 的正方形棋盘，上面填满了整数。玩家轮流走棋。第一位玩家的行动是划掉所选的一行中尚未被划掉的行。第二位玩家的行动是划掉所选的一列，这列还没有被划掉。每位棋手下完 $n - 1$ 步棋后，正好有一个整数没有被划掉。第一个玩家的目标是最大化这个数字，而第二个玩家的目标是最小化这个数字。

给定棋盘，确定游戏结束时一个未交叉数字的值。由于两位棋手都是天才，他们的下法都是最优的。

输入

输入的第一行包含一个整数 n ($2 \leq n \leq 50$)，表示棋盘的大小。随后的 n 行输入描述了棋盘的行：第 i 行包含 n 个整数 $a_{i1}, a_{i2}, \dots, a_{in}$ ($1 \leq a_{ij} \leq 2500$)，代表第 i 行的数字。

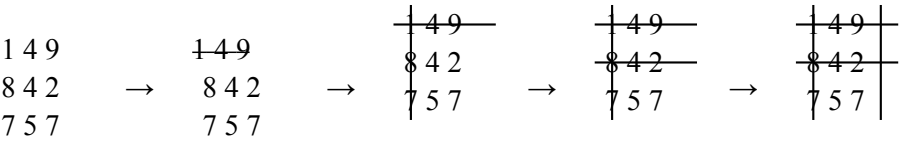
输出

输出一个整数，即游戏结束时，双方都以最佳方式进行游戏时一个未交叉数字的值。

示例

标准输入	标准输出
3 1 4 9 8 4 2 7 5 7	5

备注





水管工亨利的问题

时间限制

1 秒 内存限制

1024 兆字节

Henryk 必须完成一定数量的三维水力指令。每个订单都包括连接进水口和出水口。这种连接是通过一系列管道和弯头配件实现的。管道是直管段，弯头以直角连接管道的两端。在进水口，我们必须按照指定的进水方向放置第一个弯头，而在出水口，我们必须按照指定的出水方向放置最后一个弯头。

入口和出口都位于**水平**方向，这意味着这些方向的矢量 z 坐标为零。

为简单起见，我们假设进水口、出水口和所有弯头的尺寸为零，即它们是三维空间 R^3 中的点，管道的宽度为零，即它们是三维空间 R^3 中的段。

进水口和出水口位于不同位置。第一根管道必须从进水点开始，并与进水方向垂直。随后的每根管道都必须从前一根管道的末端开始，并与之垂直。最后一根管道的终点必须是出水点，并与出水方向垂直。每根管道的长度必须为正数。

除上述位置外，管道不能在任何其他位置相交。特别是，所有弯头必须位于不同的位置，且不能位于任何管道内，除弯头外，管道不能共享公共点。

对于每个订单，决定是否可以完成，如果可以，计算完成订单所需的最少肘数。

输入

输入的第一行包含订单数 t ($1 \leq t \leq 100$)。每个订单分四行描述。

在每个命令的第一行，有三个整数 x_1, y_1, z_1 ($-20 \leq x_1, y_1, z_1 \leq 20$)，描述了进水口的点 (x_1, y_1, z_1) 。

命令的第二行包含 2 个整数 p_1, q_1 ($-20 \leq p_1, q_1 \leq 20, (p_1, q_1) \neq (0, 0)$)，描述了进水口的方向。

接下来的两行以同样的格式描述了坐标 (x_2, y_2, z_2) 和出水口方向 (p_2, q_2) 。

进水口和出水口位于不同点： $(x_1, y_1, z_1) \neq (x_2, y_2, z_2)$ 。

水流沿矢量 $(p_1, q_1, 0)$ 水平流动，即从点 $(x_1 - p_1, y_1 - q_1, z_1)$ 方向流出。流出的水也是水平的，沿矢量 $(p_2, q_2, 0)$ 流出，即沿点的方向 $(x_2 + p_2, y_2 + q_2, z_2)$ 流出。

这些矢量的长度并不重要，重要的是它们的方向和方位。方向与水流方向一致，即朝向进水口的弯头，远离出水口的弯头。



输出

输出结果应由 t 行组成。在 第 i 行中，您应提供完成订单所需的最少弯头数量，如果订单不可行，则提供 NIE 字样。

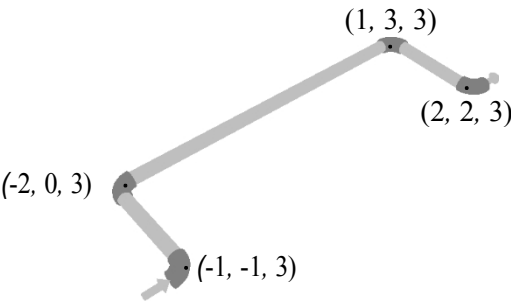


示例

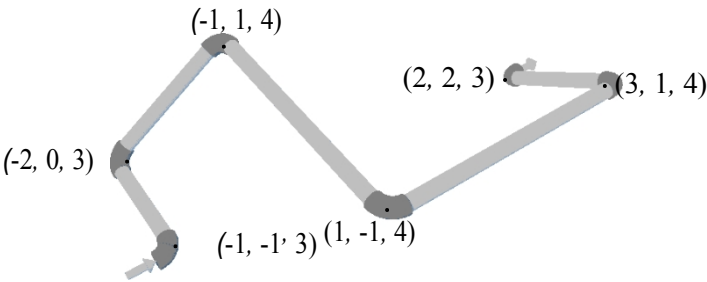
标准输入	标准输出
2	4
-1 -1 3	3
1 1	
2 2 3	
2 2	
5 5 1	
3 0	
7 6 -2	
1 -2	

备注

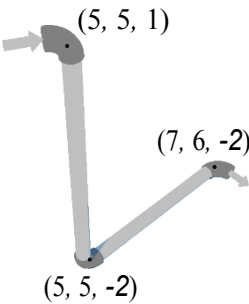
在第一个例子中，最佳解决方案是使用四个弯头，例如，依次在以下各点使用弯头 $(-1, -1, 3)$, $(-2, 0, 3)$, $(1, 3, 3)$, $(2, 2, 3)$:



一个低效的解决方案是在点 $(-1, -1, 3)$ 、 $(-2, 0, 3)$ 、 $(-1, 1, 4)$ 、 $(1, -1, 4)$ 、 $(3, 1, 4)$ 、 $(2, 2, 3)$ 使用六个弯头:



在第二个例子中，我们可以在点 $(5, 5, 1)$ 、 $(5, 5, -2)$ 、 $(7, 6, -2)$ 使用三个肘部:





问题 I. ICPC 推断

时间限制 3 秒 内存限制
1024 兆字节

让我们考虑一下规则与 AMPPZ 几乎相同的 ICPC 竞赛。参赛队提交解决方案，并对解决方案的正确性和错误性进行评估。首次提交正确解题方案会增加参赛队的时间惩罚，即从比赛开始起，⁴，每提交一次错误解题方案，⁵，则增加 **20** 分钟。第一个正确问题之后提交的所有问题都将被忽略，不影响团队得分。参赛队按正确解题的数量降序排名。出现并列时，按总罚分时间从多到少排序，再出现并列时则随机排序。

比赛刚刚结束！比赛持续了 L 分钟， D 队共提交了 N 份答卷，解决了主办方准备的 **13** 个问题。得分前三名的队伍将登上领奖台。评审团决定不破坏惊喜，不看提交的结果，甚至不看提交的是哪个问题。他们只知道每份材料的时间和提交材料的团队。评审团现在想知道领奖台会是什么样子，即排名前三的团队的顺序，每个团队都至少解决了一个问题。

从评委的角度找出可能登上领奖台的人数。我们忽略少于三个团队解决任何问题的情况。

输入

输入的第一行包含三个整数： N 、 D 和 L ($3 \leq N、D、L \leq 200\,000$)。这三个整数分别代表参赛作品数量、参赛队伍数量和比赛持续时间（以分钟为单位）。

以下 N 行中，每行有两个整数： d_i 和 t_i ($1 \leq d_i \leq D, 1 \leq t_i \leq L, t_i \leq t_{i+1}$)。这些整数表示参赛队索引和第 i 个参赛作品从比赛开始时的时间。这些作品按时间排序。

输出

输出一个整数--领奖台上三支队伍的可能有序集合数。

实例

标准输入	标准输出
4 3 300 1 10 2 25 2 30 3 50	3
4 6 200000 6 1 6 1 1 2 2 2	4



备注

在第一个测试案例中，有 3 种可能的平台配置：

- (1, 2, 3) - 如果每个小组都正确解决了一个问题，罚时总数为 (10, 25, 50)

⁴AMPPZ 测量的提交时间精确到秒。

⁵在 AMPPZ 中，只有成功编译的提交才会增加时间惩罚。



或 (10, 30, 50) 或 (10, 50, 50)。最后一个选项假设第 2 小组对同一问题先提交了错误的解法，然后又提交了正确的解法，因此罚时总数为 $30 + 20 = 50$ 。

- (1, 3, 2) --同样，总罚分时间为 (10, 50, 50)，决胜局随机决定有利于第 3 队。
- (2, 1, 3) --如果第 2 小组正确解决了两个问题，而其他小组各解决了一个问题。

在第二个样本测试案例中，可能的平台配置为 (1, 2, 6)、(2, 1, 6)、(6, 1, 2) 和 (1, 2, 6)。
(6, 2, 1)。



问题 J. 朱丽叶统一一个人

时间限制

2 秒 内存限制

1024 兆字节

如果一个二进制字符串（由 1 和 0 组成）的所有 1 构成一个连续的（可能是空的）区间，而中间没有任何 0，我们就称该字符串为**统一字符串**。这类字符串的例子有 0011110、1 和 0000。然而，二进制字符串 101 和 00100011 并不是统一字符串。

朱丽叶有一个二进制字符串 S ，她愿意删除一些字符来使字符串统一。当朱丽叶删除一个字符时，剩余的字符会滑动以填补空缺。

必须从 S 中删除多少个字符才能使剩余的字符组成统一的二进制字符串？

输入

输入包含一行字符串 S ($1 \leq |S| \leq 50$, $S_i = '0'$ 或 $S_i = '1'$)。

输出

输出一个整数 - 删除字符的最小可能数目。

示例

标准输入	标准输出
00011011001	2

备注

在字符串 00011011001 中，朱丽叶可以删除两个下划线字符，得到统一字符串 000111100。



问题 K. 路由 K 代码

时间限制 1 秒 内存限制 1024 兆字节

在 Bajtex 公司的内部网络中，有 n 个路由器，编号从 1 到 n 。每个链路连接两个不同的路由器，任意两个路由器之间最多只有一个直接链路。每两个路由器之间至少有一个链路序列相连。

系统管理员正在部署一种新的数据包路由算法。这需要为每个路由器 a 分配一个唯一的 32 位路由代码 $K(a)$ ($0 \leq K(a) < 2^{32}$ ， $K(a) \neq K(b)$ 为 $a \neq b$)。如果两个路由器通过直接链接相连，则其中一个代码应为另一个代码的一半，并向下舍入。换句话说，如果路由器 a 和 b 由一条链路连接，则

$$K(a) = \frac{K(b)}{2} \quad \text{或} \quad K(b) = \frac{K(a)}{2}$$

检查是否可以根据这些要求分配路由代码。如果可以，请计算这些代码的最小可能总和， $\sum_{i=1}^n K(i)$ 。

输入

输入的第一行包含两个整数 n 和 m ($1 \leq n \leq 200\,000$ ， $n - 1 \leq m \leq 200\,000$)，分别代表路由器数量和链路数量。

接下来的 m 行描述链接。其中第 i 行有两个整数， a_i 和 b_i ($1 \leq a_i, b_i \leq n, a_i \neq b_i$)，描述了路由器 a_i 和 b_i 之间的链接。每个链接在输入中最多出现一次（如果 $i \neq j$ ，则 $(a_i, b_i) \neq (a_j, b_j)$ 和 $(a_i, b_i) \neq (b_j, a_j)$ ）。

输出

如果可以正确分配唯一的路由代码，输出结果应包含一个整数：“.....”。
 $\sum_n K(i)$ 的最小可能总和。如果不可能，输出应只包括词 NIE。

实例

标准输入	标准输出
4 3 1 2 1 3 1 4	6



4 6	NIE
1 2	
2 3	
3 4	
4 1	
1 3	
2 4	

备注

在第一个例子中，路由器可以分别分配代码：1、0、2、3，总计 6。在第二个示例中，无法分配代码。



问题 L.随机数

时间限制 2 秒 内存限制
1024 兆字节

从 1 到 n 的数字**随机**排列。换句话说，从 1 到 n 的每个数字都正好出现一次，而且它们的顺序是随机的。

我们正在寻找**有趣的**区间，即区间内元素之和等于其长度平方的区间。形式上，在序列 a_1, a_2, \dots, a_n 中，一个有趣的区间对应于索引范围 $[p, q]$ ($1 \leq p \leq q \leq n$)，这样：

$$\sum_{i=p}^q a_i = (q - p + 1)^2$$

计算有趣区间的数量。

输入

输入的第一行包含一个整数 t ($1 \leq t \leq 200,000$)，表示测试用例的数量。每个测试用例分两行描述。

每个测试用例的第一行包含一个整数 n ($1 \leq n \leq 200,000$)，表示序列的长度。

每个测试用例的第二行包含 n 个不同的整数 a_1, a_2, \dots, a_n ($1 \leq a_i \leq n, a_i \neq a_j$ for $i \neq j$)。序列是随机选择的，这意味着 n 个序列中每个序列被选中的概率相等，不同的测试案例也是如此。不过，组织者可以在每个测试案例中任意选择数字 t 和数字 n 。

所有测试案例的 n 之和不超过 200 000。

输出

输出结果应由 t 行组成。第 i 行应包含一个整数，即第 i 个测试用例中有趣区间的数量。

示例

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

备注

在第一个测试案例中，感兴趣的区间是 $[2, 2]$ （因为 $1 = 1^2$ ）和 $[2, 3]$ （因为 $1 + 3 = 2^2$ ）。在



第二个测试案例中，感兴趣的区间是 $[1, 3]$ （因为 $3 + 4 + 2 = 3^2$ ）和 $[5, 5]$ （因为 $1 = 1^2$ ）。



问题 M. 数学锦标赛

时间限制 1 秒 内存限制 1024 兆字节

在数学锦标赛中，有两个 n 数学家，每个人的初始名气值都是 i ，可能是负数。当两位数学家进行数学对决时，组织者会以一种只有他们自己知道的方式决定胜负。失败者的名气保持不变，但决斗获胜者的名气会随着失败者名气的增加而增加 ($a[i] += a[j]$)。请注意，这甚至会导致获胜者的名气下降！

锦标赛由 n 个阶段组成。在每个阶段，主办方将参赛者配对。在每对参赛者中进行数学对决，胜者进入下一阶段，败者被淘汰。第一阶段结束后， $n-1$ ，第二阶段结束后， $n-2$ ，以此类推。最后，第 n 个阶段结束后，只剩下一名参赛者，他们将象征性地获得一块巧克力。

冠军赛结束后，计划对其中一名参赛者（不一定是巧克力持有者）进行面试。采访的最佳人选是 n 数学家中最终名气最大的两位。冠军赛的声誉和明年能否吸引到赞助商都取决于这次采访。帮助组织者匹配每个阶段的参赛者并选出优胜者，使其中一位数学家的最终名气最大化。找出这个最大值。

输入

输入的第一行包含一个整数 n ($1 \leq n \leq 16$)。

第二行包含 2 个 n 整数 a_1, a_2, \dots, a_n 。这些数字中的第 i 个数字是" i"的初始名称。
 i ($-10^6 \leq a_i \leq 10^6$)。

输出

输出结果应包含一个整数，即冠军赛结束后 n 2 位数学家之一的最大名气。

示例

标准输入	标准输出
2 5 -1 2 -10	7

备注

在第一阶段，组织者创建了以下几对数学家（这不一定是唯一的最佳选择）：

- 在有名望的一对中（一个 $i_1 = 5$ ，一个 $i_3 = 2$ ），让数学家 3 获胜；他的名望从 2 变为 $2+5 = 7$ 。参赛者 1 以 5 分被淘汰。



- 在一对 ($\gamma_2 = -1$, $\gamma_4 = -10$) 中，让数学家 2 获胜；他的名气从 -1 变为 $-1 + (-10) = -11$. 参赛者 4 以 -10 分被淘汰。

在这种情况下，名气分别为 7 和 -11 的数学家进入第二阶段。3 号选手以 7 分的成绩被淘汰，2 号选手获得巧克力，最终成绩为 -4 分。

四位数学家的最终名气分别为 5、-4、7、-10。名气最高的是 7，这是可能的最大值。