

第三届环球杯



第五赛段莫斯科

2024 年 7 月 27-28 日

这套试题应包含 13 个问题，共 21 页。



问题 A.计数排列

27-28 日

输入文件：标准输入
输出文件：标准输出
时间限制 1 秒 内存限制 512 兆字节

给你一个大小为 n 的数组 a 和两个整数 m_1 和 m_2 。如果满足以下条件，我们称数组 p 为好数组：

- p 是大小为 n 的排列，由从 1 到 n 的不同整数组成。
- 数组 $a_{p_1} \bmod m_1, a_{p_2} \bmod m_1, \dots, a_{p_n} \bmod m_1$ 是一个非递减数组。
- 数组 $a_{p_1} \bmod m_2, a_{p_2} \bmod m_2, \dots, a_{p_n} \bmod m_2$ 是一个非递增数组。

计算以 998244353 为模数的好排列 p 的个数。

输入

输入由多个测试用例组成。第一行包含一个整数 t ($1 \leq t \leq 10\,000$) --测试用例数。测试用例说明如下。

每个测试用例的第一行包含三个整数 n, m_1 和 m_2 ($1 \leq n \leq 100\,000, 1 \leq m_1, m_2 \leq 10^4$) --数组的长度和两个模块。

下一行包含 n 个整数 a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$) - 数组。保证所有测试用例的 n 之和不超过 100 000。

输出

对于每个测试用例，打印一个整数--问题答案的模数 998244353。

示例

标准输入	标准输出
3	2
5 2 3	0
1 2 3 4 10	6
4 2 4	
1 2 3 4	
3 8 9	
1 1 1	

备注



在第一个测试案例中，有两种好的排列组合 $p = [4, 5, 1, 3]$ 和 $p = [2, 5, 4, 1, 3]$ 。在

第二个测试案例中，没有好的排列组合。

在第三个测试案例中，所有 6 种排列方式都是好的。



问题 B. 书架跟踪

27-28 日

输入文件：标准输入

输出文件：标准输出

时间限制 2 秒 内存限制

512 兆字节

编程历史与哲学 (PHP) 系主任 E.D. Pryanik 教授坚信，只有彻底研读了他的基础著作《编程科学》全部 N 卷的学生才能在他的指导下撰写学期论文。学生必须研读所有书籍，然后 E.D. Pryanik 才会对他进行面试。如果教授对结果感到满意，就会为学生指定学期论文的题目。

教授想让你帮忙追踪他办公室书架上书籍的顺序。书架上有 n 本书，编号为 1 到 n 的整数。书架可以表示为从 1 到 n 的整数的排列 - 书架上书籍的顺序。

书架可以进行两种操作：

- 教授调换了书架上的两本书。
- 一名新生来到办公室，要求读完 n 本书。他按照从

1 到 n 的顺序阅读书籍：

- n 次，学生拿下一本书。书的位置上保留一个空位。
- 如果空格左边的书本数量小于空格右边的书本数量，他就把空格左边的所有书本向右移动一个位置。否则，他将空格右边的所有书本向左移动一个位置。结果，空位移动到第一个位置或最后一个位置。
- 学生将书放回空白处。

给出书架上书籍的初始顺序和运算顺序。请告诉教授所有运算后书籍的最终顺序。

输入

第一行包含两个整数 n, q ($2 \leq n, q \leq 3 \cdot 10^5$)。保证 n 是偶数。

第二行包含从 1 到 n 的数字排列，即书架上书籍的初始顺序。

下面每 q 行包含下一个操作的说明。每行以符号 "E" 或 "R" 开头，说明操作的类型。

- 如果符号为 "E"，则该行包含两个整数 i, j ($1 \leq i, j \leq n, i \neq j$) --应交换的图书的索引 (图书编号，而不是排列中的位置)。



- 如果符号为 "R"，则学生必须到办公室阅读所有书籍。

输出

打印 n 个数字 - 书架上书籍的最终顺序。



示例

27-28 日

标准输入	标准输出
8 4 7 2 6 1 8 3 5 4 R E 2 3 R E 1 6	7 1 3 6 2 4 5 8



问题 C. 粉刷围栏

27-28 日

输入文件：标准输入
输出文件：标准输出
时间限制：2 秒 内存限制：512 兆字节

在 ICPC 比赛中大败之后，Matvey 决定去刷围栏。
他的任务是把一个 "小"（对他来说）栅栏涂成黑色。栅栏是一个 $n \times m$ 大小的方格矩形。初始颜色已给出。
马特维可以通过以下操作进行绘画：

- 选择一些水平或垂直直线，沿着边缘与矩形相交。
- 在所选线条的一侧，不要更改颜色。
- 在与所选线相对的另一侧，首先将所有单元格涂成白色。然后，在这一边，将与所选线对称的单元格涂成黑色的所有单元格涂成黑色。

要把所有单元格涂成黑色，最少要进行多少次操作？

输入

第一行包含两个整数 n, m ($1 \leq n \cdot m \leq 10^6$) -- 栅栏的大小。
接下来的 n 行中，每一行都包含一个由 m 个符号 0 或 1 组成的字符串。如果第 i 行字符串中的第 j 个符号等于 0，则第 i 行和第 j 列中的单元格为白色。否则为黑色。
保证至少有一个黑格存在。

输出

打印一个整数 - 用于将所有单元格涂黑的最小描述操作数。

示例

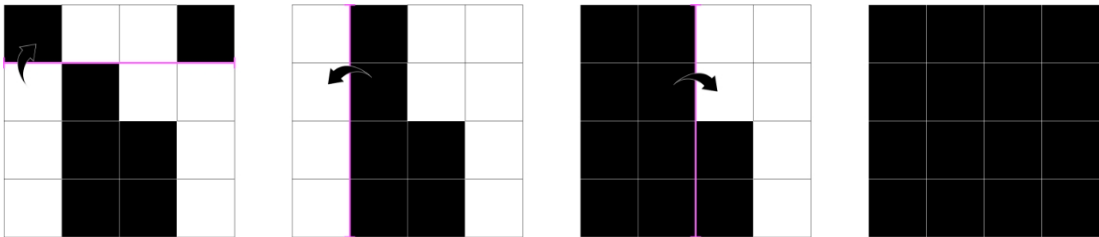
标准输入	标准输出
4 4 1001 0100 0110 0110	3



备注

27-28 日

在第一次测试中，将所有单元格涂成黑色的方法如下图所示。





问题 D. 有许多最大值的函数^{27-28 日}

输入文件：标准输入
输出文件：标准输出
时间限制：1 秒 内存限制：512 兆字节

对于正整数 $a > 0$ ，让我们定义函数 $f_a: \mathbb{Z}_{\geq 0} \rightarrow \mathbb{Z}$ ：

$$f_a(x) = \begin{cases} a + x, & \text{如果 } x \leq a \\ 0, & \text{否则} \end{cases}$$

给你一个正整数 b 。构造由不同整数 a_1, a_2, \dots 组成的非空集。 a_n ，这样

函数 $f(x) = \max_{i=1}^n f_{a_i}(x)$ 至少有 b 具有最大函数值的点。形式上

$$x \in \mathbb{Z}_{\geq 0} : f(x) = \max_{y \in \mathbb{Z}_{\geq 0}} f(y) \geq b$$

输入

唯一一行包含一个整数 b ($1 \leq b \leq 100\,000$)。

输出

在第一行，打印一个整数 n ($1 \leq n \leq 500\,000$) -- 数组 a 的大小。在下一行

， n 个不同的整数 a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^{12}$)。

保证在给定的约束条件下答案是存在的。如果有多个答案

，可以打印任意一个。

示例

标准输入	标准输出
4	5 2 3 5 10 12

备注

在第一次测试中， $f(x) = f_2(x) + f_3(x) + f_5(x) + f_{10}(x) + f_{12}(x)$ 。最大值为 $\max_{y \in \mathbb{Z}_{\geq 0}} f(y) = 42$ 。我们

有 4 个点，最大值为 $f(2) = f(3) = f(5) = f(10) = 42$ 。



问题 E. 修建围栏

27-28 日

输入文件：标准输入
输出文件：标准输出
时间限制：1 秒 内存限制：512 兆字节

尼科季姆家的院子里最近布置了一个花坛。邻居家的狗经常从花坛里跑过，尼科季姆决定保护这些花。他打算在花坛周围建一个小栅栏。

在 "All for garden "商店，您可以买到一段塑料栅栏。每段长度为 l 。
购买后，如果顾客愿意，销售人员可以将一些初始部分切割成任意大小的两个小部分（长度不一定是整数）。新的部分不能再次切割。不同的初始部分可以切割成不同长度的新部分。
花坛是一个大小为 $w \times h$ 的矩形，尼科季姆想做一个正好是这个大小的栅栏。为此，矩形的每一边都应表示为一些截面的总和（每个截面最多使用一次）。请注意，不一定要使用截面的所有部分。
确定他应该购买的最小长度段数。

输入

第一行包含一个整数 t ($1 \leq t \leq 10^4$) --测试用例数。
接下来的 t 行中，每一行都包含三个整数 w, h, s ($1 \leq w, h, s \leq 10^8$) --花坛的尺寸和路段的长度。

输出

针对每个测试案例，打印 Nikodim 建造栅栏所需的最小长度 s 段数。

示例

标准输入	标准输出
7	8
7 9 4	2
1 1 2	4
1 1 4	10
4 6 2	4
3 3 5	8
10 6 4	5
1 11 5	

备注



在第一个测试案例中，应购买 8 个分段。之后，可以将两段分成长度为 1 和 3 的小段。然后，可以建造栅栏：2 乘以 $7 = 4 + 3$ 和 2 乘以 $9 = 4 + 4 + 1$ 。



问题 F. 传送门

27-28 日

输入文件：标准输入
输出文件：标准输出
时间限制：1 秒 内存限制：512 兆字节

其中一个箱子里有宝藏，其余的箱子都是空的。你想找到宝藏，但打开所有的箱子需要太多时间，所以你想确定哪个箱子里有宝藏。

您还拥有 n 个传送阵，其中第 i 个位于第 i 个箱子的顶部。您可以激活任意一个传送阵，宝藏会被对称地传送到传送阵的另一侧。因此，如果您激活了传送门 a ，而宝藏位于箱子 b 内，那么宝藏的新位置将是 $b + (a - b) - 2$ 。如果有编号为 $b + (a - b) - 2$ 的箱子，宝藏就会被传送到编号为 $b + (a - b) - 2$ 的箱子。如果没有该数字的箱子，宝藏将留在箱子 b 中，您将知道传送没有成功。

每个传送门都有自己的使用成本，使用一次传送门 i 需要支付 c_i 。找出你需要花费的最小金额，这样在使用多个传送门后，你就可以确定宝藏的位置了。

输入

第一行包含一个整数 n ($1 \leq n \leq 500$) --箱子的数量。
第二行包含 n 个整数 c_1, c_2, \dots, c_n ($1 \leq c_i \leq 10^9$) --每个传送门的单次使用成本。

输出

输出您需要的最低金额，以便确定宝藏的位置。

实例

标准输入	标准输出
3 5 2 1	4
12 18 19 11 2 20 15 18 1 14 1 1 1	6

备注

在第一个例子中，首先您可以启动传送 3。如果传送成功，唯一有效的箱子就是 3。然后您就可以启动传送门 2，之后宝藏就会出现在 2 号或 3 号箱子中。之后您可以激活传送 3，如果传送成功，宝藏就在 3 号箱子里，否则就在 2 号箱子里。



问题 G. 指数计算器

27-28 日

输入文件：标准输入

输出文件：标准输出

时间限制 1 秒 内存限制

512 兆字节

Kirill 想要制作自己的计算器。然而，他面临着一个严重的问题--他应该添加计算一些复杂函数的按钮。

他的计算器有 16 个寄存器 $\$0$ 、 $\$1$ 、...、 $\$15$ ，每个寄存器包含 64 位浮点数（双精度浮点格式）。

计算器的操作应通过使用寄存器的命令序列来表示：

- $\$i = \text{操作数1 操作数2}$
- $\$i = \text{值}$

在第一条命令中，操作数 1 和操作数 2 可以是实数或寄存器之一。操作数中至少有一个应是寄存器，操作应为 $+$ 、 $-$ 、 $*$ 中的一种，分别对应于和、减、乘。

在第二条命令中，值应该是某个实数（不允许使用寄存器）。

计算该操作时，寄存器 1 的初始值等于 x （操作的输入值），其他寄存器的初始值等于 0。首先，计算右侧的值，并将其分配给左侧的寄存器。操作的输出是寄存器 $\$0$ 的值，其他寄存器的值不予考虑。

编写一个最多包含 25 个命令的序列，以较高的精度计算函数 e^x ，适用于所有 $x \in [-20, 20]$ （ e 是欧拉数和自然对数的底数）。

输入

你可以把这个问题看作是一个纯输出问题。

问题中只有一个测试，包含一个字符串 "input"。

输出

第一行包含一个整数 k ($0 \leq k \leq 25$) --命令的数目。接下来的 k 行应描述一连串能精确计算 e^x 的命令。

接下来的 k 行中，每一行都应包含一条命令，格式为两种给定格式之一。应满足上述条件。对于所有寄存器 $\$i$ ，应满足 $0 \leq i \leq 15$ 的条件。



请注意，在这道题中，重要的是第一行只打印命令的数目，接下来的 k 行中每行打印一个操作（中间不能有空行）。所有操作数、运算符和 $=$ 符号都应用空格隔开。

对于整数 n_{05} ($-2 \cdot 10^6 \leq n \leq 2 \cdot 10^6$) 的所有 $x = n$ ，将执行命令序列。假设答案等于 $y_1 = e^x$ ，计算出的答案等于 y_2 。答案将被视为如果 $\frac{|y_1 - y_2|}{y_1} \leq 10^{-9}$ ，则正确。



示例

27-28 日

标准输入	标准输出
输入	6 \$0 = \$0 * \$0 \$0 = 2.718281828459 \$0 = \$1 + \$0 \$15 = 8.1000000737 * \$0 \$12 = \$15 * 0.123456789 \$0 = \$12 - \$1

备注

所提供的答案只是一个例子，是不正确的。

不难看出，对于所有 x ，它都能近似计算出 e （因此对于 $x = 1$ ，答案是正确的）。



问题 H. 古国

27-28 日

输入文件：	标准输入
输出文件：	标准输出
时间限制	10 秒 内存限
制	512 兆字节

在古代世界，战争频仍。人们认为，要保护一座城市，就必须用形成凸多边形的筑城墙将其围起来。某个古代国家的国王决定建造一些城市，并用防御墙保护每座城市。请帮助他实现对国家的最高保护。

古国的领土是一个简单的多边形 $P P_{i_2} \dots P_n$ 。严格位于该多边形内部或边界上的所有点都受该国控制。多边形的所有顶点都是不同的，多边形中没有两条边相交或相触，只有连续的边在它们的共同顶点相触。每两条连续的边都不相交。

在多边形的每个点 P_i 都有一座塔。

最初，国家没有城市。国王可以建造一座城市 $P P_{i_1 i_2} \dots P_{i_k}$ ：

- 点 $P P_{i_1 i_2} \dots P_{i_k}$ 是构成一个国家的多边形的不同顶点。
- 多边形 $P P_{i_1 i_2} \dots P_{i_k}$ 是凸形，面积为正。
- 点 $P_{i_1}, P_{i_2}, \dots, P_{i_k}$ 按此顺序出现在城市边界的逆时针方向上。
- 除了 P_{i_1}, P_{i_2}, \dots 以外，城市边界上没有其他塔楼 P_{i_0}, P_{i_k} 。
- 城市边界内或边界上的所有点都受国家控制，即多边形 $P P_{i_1 i_2} \dots P_{i_k}$ 边界内或边界上的所有点 $\dots P_{i_k}$ 位于多边形内部或边界上 $P P_{i_2} \dots P_n$ 。

给您两个非负整数 w 和 c 。一个城市的保护级别等于 $2 \cdot \text{area}(P P_{i_1 i_2} \dots P_{i_k}) + w \cdot k + c$ 。国家的保护级别是其所有城市保护级别的总和。

国王希望在全国范围内选择一些城市，使两座城市没有共同点（包括边界）。请注意，国家的某些点可能不在任何城市的覆盖范围内。

国家可能的最高保护级别是多少？

输入

第一行包含一个整数 n ($3 \leq n \leq 200$) --国家的顶点数。

接下来的 n 行中，每一行都包含两个整数 x_i, y_i ($|x_i|, |y_i| \leq 10^6$) - 点 P_i 的坐标。保证顶点按逆时针顺序给出。



下一行包含两个整数 w, c ($0 \leq w, c \leq 10^{13}$) 日

输出

打印一个整数--国家可能的最高保护级别。



实例

27-28 日

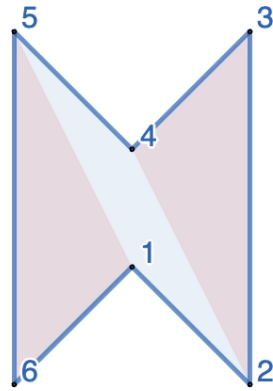
标准输入	标准输出
6 0 0 1 -1 1 2 0 1 -1 2 -1 -1 1 2	16
12 0 0 1 1 2 0 4 1 5 0 5 1 8 0 9 1 10 0 11 1 5 5 1 4 3 1000000	3000063
12 0 0 1 1 2 0 4 1 5 0 5 1 8 0 9 1 10 0 11 1 5 5 1 4 0 9	61

备注

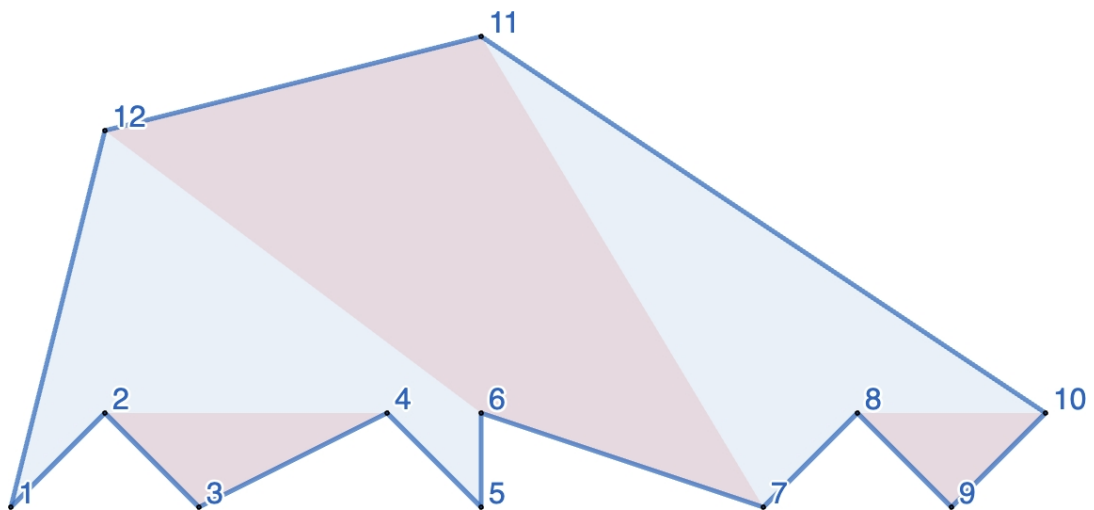
下面的图片显示了前三个示例的城市（城市用红色标出）。



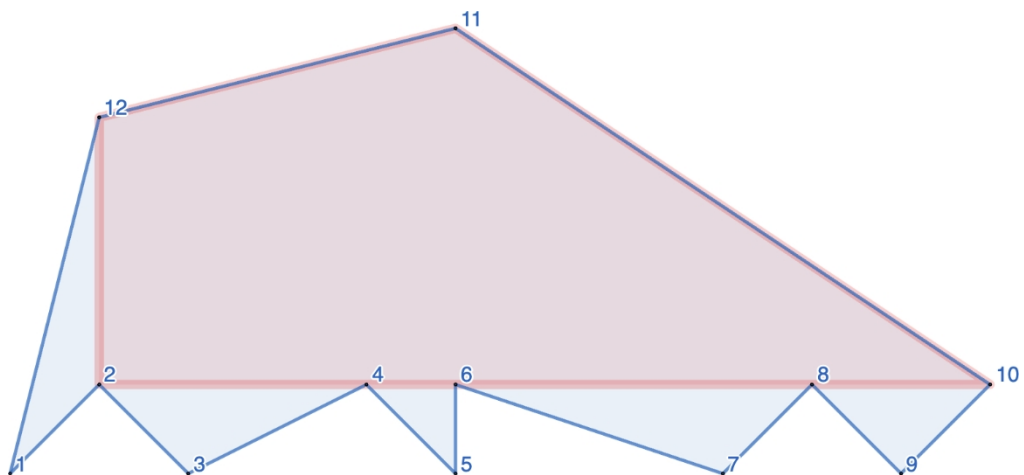
27-28 日



示例 1（可能的最佳答案：{1, 5, 6}, {2, 3, 4}）



示例 2（可能的最佳答案：{2, 3, 4}, {6, 7, 11, 12}, {8, 9, 10}）



示例 3（可能的最佳答案：{2, 4, 6, 8, 10, 11, 12}）



问题 1. 马克笔总和

27-28 日

输入文件：	标准输入
输出文件：	标准输出
时间限制	1 秒 内存限制
	512 兆字节

这是一个运行两次的问题。您的解决方案将被执行两次。

伊万学习很差。在学习期间，他只得了 1 分和 2 分。为了向父母汇报自己的成绩，他决定用分数总和的方式（这样父母就不会看出他的分数很差）。

Ivan 的标记可以用字符串 s 表示，该字符串由字符 1 和 2 组成。他开始下载，但不幸的是，网络连接中断了，他只下载了字符串 $s[1..i]$ 的前缀（对于某些 $1 \leq i < |s|$ ，所以这个前缀不是空的，也不等于完整的字符串）。重新连接后，将下载字符串的剩余后缀。

您的解决方案将被执行两次：

1. 在第一次执行时，您将得到下载的前缀： $s[1..i]$ 。您应该给出某个 $0 \leq d < \min(i + 1, 2000)$ 和 $0 \leq info < 2000$ 的数字。
 2. 在第二次执行时，您将得到数字 $info$ 和字符串的剩余后缀，以及下载的前缀的最后 d 个符号： $s[(i + 1 - d) .. |s|]$ 。
- 您应该给出 s 中所有符号的总和。

输入

第一行包含一个整数类型 ($1 \leq type \leq 2$)。第一次执行时， $type = 1$ ，第二次执行时， $type = 2$ 。

输入由多个独立的测试用例组成。第二行包含一个整数 t ($1 \leq t \leq 1000$) - 测试用例数。测试用例说明如下。

- 如果 $type = 1$ ，接下来的 t 行都包含一个字符串，由字符 1 和字符 2 - 下载的前缀 $s[1..i]$ 组成。
- 如果 $type = 2$ ，接下来的 t 行中的每一行都包含一个整数 $info$ ($0 \leq info < 2000$)，表示您的解决方案是在此测试用例的第一次执行中给出的，以及一个字符串，由字符 1 和字符 2 组成 - 下载的后缀 $s[(i + 1 - d) .. |s|]$ 表示 d ，表示您的解决方案是在此测试用例的第一次执行中给出的。



保证所有字符串 s 和数字 i 都是事先固定的。保证所有测试用例的 $|s|$ 总和

和不超过 10^6 。

请注意，测试用例在第二次执行时可以重新排序。

输出

- 如果 $type = 1$ ，对于每个测试用例，您的解决方案应打印两个整数 $0 \leq d < \min(i + 1, 2000)$ ， $0 \leq info < 2000$ 。它们将用于该测试用例的第二次执行。
- 如果 $type = 2$ ，则对于每个测试用例，解决方案都应打印一个整数，即字符串 s 中的总和。

示例

请注意，第一次执行时给出的解决方案输出结果是示例，可能与您的解决方案不同。



第一次执行。

27-28 日

标准输入	标准输出
1	0 42
3	1 11
1	4 22
222	
1212111122	

第二次执行。

标准输入	标准输出
2	12
3	21
11 22211	3
22 11222221	
42 2	

备注

在第一个测试中，有 3 个测试用例，它们的字符串是

12
2222211
12121111222221

请注意，在第二次执行时，测试用例被重新排序。



问题 J. 前缀能被后缀整除^{27-28 日}

输入文件：标准输入

输出文件：标准输出

时间限制 8 秒 内存限制

512 兆字节

给你两个正整数 n 和 c 。

考虑一个正整数 x 及其十进制表示 $x = (x_{k-1}x_{k-2} \dots x_1x_0)_{10}$ （不含前导零）。如果数 x 的十进制表示法可分为非空的前缀 $p = (x_{k-1} \dots x_i)_{10}$ 和非空的后缀 $s = (x_{i-1} \dots x_0)_{10}$ （对于某个 $0 < i \leq k-1$ ），使得 p 能被 $s + c$ 整除，我们就称该数 x 为好数。注意， $x_{i-1} = 0$ 是可能的。

计算从 1 到 n 的好整数的个数。

输入

唯一的一行包含两个正整数 n 和 c （ $1 \leq n \leq 10^{14}$ ， $1 \leq c \leq 10^4$ ）。

输出

打印一个整数 - 从 1 到 n 的整数个数。

实例

标准输入	标准输出
20 1	2
111 4	9
1111 10	75
1000000 7	111529

备注

在第一次测试中，好的数字是 10、20。

在第二次测试中，好的数字是 40、51、62、73、80、84、95、101、106。



问题 K. 火车站

27-28 日

输入文件：标准输入
输出文件：标准输出
时间限制：2 秒 内存限制：512 兆字节

您是火车站的站长。火车站有 n 个铁路道岔，道岔之间有 $n - 1$ 条铁轨，因此所有道岔都由一个铁路网连接。第 i 条铁轨连接道岔 a_i 和道岔 b_i ，长度为 c_i 米。

有 m 辆列车要停放在车库。第 i 辆列车将从开关 1 进入车库，并以最短路径驶向开关 s_i ，直到它的第一个车皮到达开关 s_i 。所有其他车皮应按连续顺序依次驶入。

不幸的是，每段铁轨只能容纳一列火车，而且车库内可能装不下所有火车。已知第 i 列火车有 k_i 个车皮，编号从 1 到 k_i 。第 j 个车皮的值为 $v_{i,j}$ ，长度为 $l_{i,j}$ 米。当车皮在车厂内时，它占用了 $l_{i,j}$ 米的铁路（可能有多条轨道）。对每列火车而言，只有部分（可能是零，也可能是全部）第一节车皮可以进入车库。

车厂。如果第 i 列火车的前 t_i 个车皮进入车厂，它们将连续占用 $\sum_{j=1}^{t_i} l_{i,j}$ 米向 1 号道岔方向的轨道。请再次注意，所有进入车库的车厢都应在车库内，两节车厢可以相碰，但不能占用同一段铁轨。

您可以选择火车进入车库的顺序，也可以选择每列火车有多少节头等车皮进入车库。求车库内可容纳的所有车皮的最大总价值。

输入

第一行包含两个整数 n 和 m ($2 \leq n \leq 200\,000$, $1 \leq m \leq 200\,000$) -- 车厂的开关数和列车数。

接下来的 $n-1$ 行描述的是铁轨。每行包含三个整数 a_i , b_i , 和 c_i ($1 \leq a_i, b_i \leq n$, $1 \leq c_i \leq 10^9$) - 铁路轨道连接的开关和轨道长度。

接下来的 $3m$ 行描述了列车。对于每列火车，第一行包含两个整数 k_i 和 s_i ($1 \leq k_i \leq 200\,000$, $1 \leq s_i \leq n$) -- 火车的车厢数和火车将停靠的开关。

第二行包含 k_i 个整数 $v_{i,1}, v_{i,2}, \dots, v_{i,k_i}$ ($1 \leq v_{i,j} \leq 10^9$) - 第 i 列火车的车皮值。

第三行包含 k_i 个整数 $l_{i,1}, l_{i,2}, \dots, l_{i,k_i}$ ($1 \leq l_{i,j} \leq 10^9$) - 第 i 节车厢的长度

火车
保证 $\sum k_i \leq 200\,000$ 。

输出



输出车库内可容纳的所有车厢的最大总价值。



实例

27-28 日

标准输入	标准输出
4 2 1 2 2 3 2 1 2 4 2 3 3 1 1 1 1 1 1 1 4 3 3	4
6 4 1 2 2 2 3 1 2 4 2 4 5 1 4 6 2 2 3 1 1 2 1 1 5 3 2 1 4 5 4 3 6 1 1 10 1 2 2	12



问题 L. 阵列展开

27-28 日

输入文件：标准输入

输出文件：标准输出

时间限制 3 秒 内存限制

512 兆字节

Vasya 想要构造一个由非负实数组成的大小为 n 的数组 a 。

他有 m 个段 $[l_1, r_1], [l_2, r_2], \dots, [l_m, r_m]$ ($1 \leq l_i \leq r_i \leq n$)，定义了数组 a 的 m 个子数

组。让我们对数组 a 的扩散定义如下：

- 计算子数组总和 $s_i = \sum_{j=l_i}^{r_i} a_j$ 。
- 数组 a 的展宽等于 $\frac{\max_{j=1}^m s_j}{\min_{j=1}^m s_j}$ 。如果最小 $s_j = 0$ ，传播等于 $+\infty$ 。

求所有可能数组 a 的最小差。答案应为 998244353。

我们可以证明，每个答案都可以写成 $\frac{p}{q}$ 的形式，其中 p 和 q 是相对质数整数

而 $q \not\equiv 0 \pmod{998244353}$ 。通过模 998244353 得到的答案等于 $(p \cdot q^{-1}) \pmod{998244353}$ 。

输入

输入由多个测试用例组成。第一行包含一个整数 t ($1 \leq t \leq 2000$) -- 测试用例数。测试用例说明如下。

每个测试用例的第一行包含两个整数 n, m ($1 \leq n \leq 10^9, 1 \leq m \leq 2000$) -- 数组大小和段数。

接下来的 m 行中的每一行都包含两个整数 l_i, r_i ($1 \leq l_i \leq r_i \leq n$) -- 第 i 段的描述。保证所有测试用例的 m 之和不超过 2000。

输出

为每个测试用例打印一个整数，即问题答案的模数 998244353。



示例

27-28 日

标准输入	标准输出
3	1
3 3	2
1 3	499122178
2 3	
1 2	
12 6	
2 3	
5 7	
1 9	
4 8	
1 2	
7 11	
4 5	
3 4	
2 3	
1 2	
4 4	
1 1	

备注

在第一个测试案例中，最小传播为 1，例如， $a = [0, 3, 0]$ 。子阵列和为 $s = [3, 3, 3]$ 。

在第二个测试案例中，最小差值为 2，例如，在下列情况下可以达到这个值
 $a = [0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 2, 0]$ 。子数组和为 $s = [1, 1, 2, 1, 1, 2]$ 。

在第三个测试案例中，最小传播范围是³，例如，在 $a = [2, 1, 1, 2]$ 的情况下就可以实现。
子数组和为 $s = [3, 2, 3, 2, 2]$ 。



问题 M. 联合变形虫

27-28 日

输入文件：标准输入
输出文件：标准输出
时间限制：1 秒 内存限制：512 兆字节

一个圆圈上有 n 个变形虫。对每 i 个变形虫

i 和 $i+1$ 是邻居。此外，变形虫 1 和 n 也是邻居。第 i 个变形虫的体积为 v_i 。

两个相邻的变形虫可以结合在一起。结果，两个阿米巴消失，一个新的阿米巴出现在它们的位置上。新变形虫的体积等于两个变形虫的体积之和。这个操作的成本等于两个变形虫的最小体积。变形虫将不断合并，直到剩下一个变形虫为止。

在剩下一个变形虫之前，所有操作的最低总成本是多少？

输入

输入由多个测试用例组成。第一行包含一个整数 t ($1 \leq t \leq 10^5$) --测试用例数。测试用例说明如下。

每个测试用例的第一行包含一个整数 n ($2 \leq n \leq 2 \cdot 10^5$) - 变形虫的数量。每个测试用例的第二行包含 n 个整数 a_1, a_2, \dots, a_n ($0 \leq a_i \leq 10^9$) - 变形虫的体积。保证所有测试用例的 n 之和不超过 $2 \cdot 10^5$ 。

输出

对于每个测试用例，打印一个整数 - 所有操作的最小总成本，直到只剩下一个变形虫。

示例

标准输入	标准输出
3	2
3	1
1 1 1	42
4	
0 1 0 2	
2	
100 42	