2023 icpc 国际大学生程序设计竞赛亚洲区域赛（济南站）

正式赛

2023 年 12 月 3 日

试题列表

A	很多很多头
B	图划分 2
C	开灯 2
D	最大数码
E	我只想要...多一个...
F	向未来说你好
G	来自知识的礼物
H	基本子串结构
I	奇怪的排序
J	计算智能
K	彩虹子数组
L	铁路环游
M	近似凸多边形

本试题册共 13 题，19 页。
如果您的试题册缺少页面，请立即通知志愿者。

由 SUA 程序设计竞赛命题组命题。
https://sua.ac/

承办方

命题方

Problem A. 很多很多头

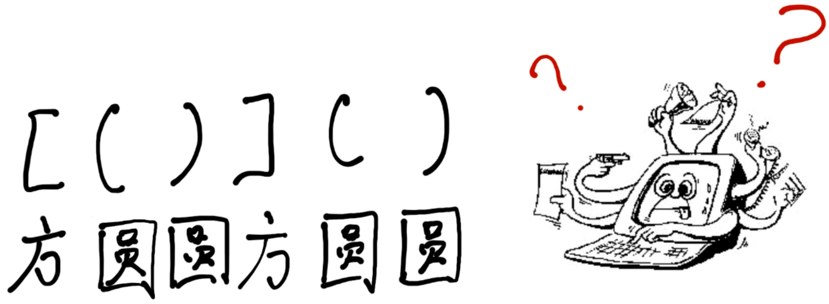
很多头杯，简称为 MHC，是一项面向拥有**很多很多头**的选手的世界级程序设计竞赛。这场赛事的裁判长小青鱼，正在考虑为每一位选手设计一个身份号码。

“那么，就是它吧，”小青鱼想，“我们来使用一些括号序列！” 他为每个选手分配了一个独一无二的平衡括号序列，每个括号序列包含两种括号——圆括号（也被称为小括号），以及方括号。为了确保您理解平衡括号序列的概念，小青鱼准备了如下平衡括号序列的形式化定义：

- ε （一个空字符串）是一个平衡括号序列。
- 如果 A 是一个平衡括号序列， 那么 (A) 和 $[A]$ 也都是平衡括号序列。
- 如果 A 和 B 是平衡括号序列，那么 AB 也是一个平衡括号序列。

例如，“()”， “[()]” 以及 “[()]()” 是平衡括号序列，但 “)(”， “[(])“ 和 “[)“ 不是。

对于我们的很多个头的选手，记住括号序列并不是一项困难的任务。然而，问题出现在了他们独特的能力上：因为他们有太多的头，因此他们无法分辨出括号的方向！结果是，与原来的平衡括号序列相比，他们记忆中的序列可能有部分括号的方向出现了改变。例如，括号序列 “[()]()” 可能会被记忆成 “]))]))” 或 “]()]))”。幸运的是，括号的种类仍然保持不变。



在比赛当天，在小青鱼收到了每位选手的括号序列后，一个问题出现了：原始的括号序列能否被唯一确定？换句话说，小青鱼需要确定给定的括号序列是否对应着唯一一个平衡括号序列。

请帮助小青鱼完成这项任务，来确保我们拥有很多个头的朋友们能够参加比赛！

Input

有多组测试数据。第一行输入一个整数 T 表示测试数据组数，对于每组测试数据：

第一行输入一个由 ‘(’， ‘)’，‘[’，以及 ‘]’ 组成的字符串 S （$1 \le |S| \le 10^5$）表示括号序列。

保证：

- 所有数据的 $|S|$ 之和不超过 10^6.
- 每个括号序列都是通过改变某个平衡括号序列的一些括号的方向所得到的。

Output

对于每组数据：

- 如果给定的括号序列可以对应不止一个平衡括号序列，输出一行 `No`。

- 否则，输出一行 `Yes`。

Example

standard input	standard output
6)) ((([()] ()[()]() ([()]) ([])([])	Yes No Yes No Yes No

Note

对于第一组样例数据，括号序列对应唯一一个平衡括号序列：`()`。所以答案是 `Yes`。

对于第二组样例数据，括号序列可以对应两个不同的平衡括号序列：`(())` 和 `()()`。所以答案是 `No`。

对于第三组样例数据，括号序列对应唯一一个平衡括号序列：`[()]`。所以答案是 `Yes`。

对于第四组样例数据，括号序列可以对应两个不同的平衡括号序列：`(([()]))` 和 `()[()]()`。所以答案是 `No`。

对于第五组样例数据，括号序列对应唯一一个平衡括号序列：`([()])`。所以答案是 `Yes`。

对于第六组样例数据，括号序列可以对应三个不同的平衡括号序列：`([])([])`, `([]()[])` 和 `([[()]])`。所以答案是 `No`。

Problem B. 图划分 2

在成功解决了《Cut Cut Cut！》这道题目之后，小青鱼想要进一步提高他在划分图的连通块方面的能力。

有一天，一位神秘的智者向小青鱼提出了一个问题。在这个问题中，小青鱼被给定了一棵由 n 个节点组成的无根树，以及一个整数 k。设 E 为树中所有边的集合，小青鱼的任务是找到一个子集 $E' \subseteq E$，使得在移除 E' 中的所有边后，图被划分为若干个连通块，且每个连通块的大小均为 k 或 $(k+1)$。

当然，作为一位分割事物的大师，小青鱼轻松地解决了这个问题。但是神秘智者的欲望远不止于此。智者不仅想要掌握事物，还想了解所有可能的结果。因此，他要求小青鱼计算出有多少种选择 $E' \subseteq E$ 的方法满足上述条件。两种方案被视为不同的，若它们选择的边的子集不相同。

请帮助小青鱼完成这一挑战。由于答案可能很大，您只需提供答案对 $998\,244\,353$ 取模后的结果。

Input

有多组测试数据。第一行输入一个整数 T 表示测试数据组数，对于每组测试数据：

第一行输入两个整数 n 和 k（$2 \le n \le 10^5$，$1 \le k \le n$）表示树的节点数量以及较小的连通块的目标大小。

对于接下来 $(n-1)$ 行，第 i 行输入两个整数 u_i 和 v_i（$1 \le u_i, v_i \le n$）表示一条连接节点 u_i 与 v_i 的边。

保证所有数据 n 之和不超过 3×10^5。

Output

每组数据输出一行一个整数，表示选择子集 E' 的方案数对 $998\,244\,353$ 取模后的结果。

Example

standard input	standard output
2 8 2 1 2 3 1 4 6 3 5 2 4 8 5 5 7 4 3 1 2 1 3 2 4	2 1

Note

令 (u,v) 表示一条连接节点 u 和 v 的边。对于第一组样例数据，两个合法的边的子集为 $\{(2,4),(3,5)\}$ 和 $\{(1,2),(3,5)\}$。

Problem C. 开灯 2

Lux et Veritas
（光明与真知）

受期待的宇宙杯（Universal Cup）决赛即将来临！小青鱼正在忙着准备比赛的场地。为了使得场地炫彩夺目，小青鱼计划悬挂一些电灯泡。

小青鱼有 m 根电线，并打算用这些电线连接 n 个灯泡。每根电线需要连接两个不同的灯泡，并使得所有灯泡形成一个连通块。为了安全起见，任意两个灯泡之间最多只能有一根电线把它们直接连起来，且每个灯泡最多只能连接 d 根电线。

连接好灯泡后，小青鱼想要点亮其中一些灯泡。考虑到亮着的灯泡会产生热量，让两个相邻的灯泡同时点亮可能会有危险。因此，如果两个灯泡之间直接通过电线连接，则它们不能同时被点亮。另一方面，他也不想让灯光太少，所以他不希望看到一个灯泡没亮，且与之直接连接的所有灯泡也都没亮。

小青鱼非常好奇在这些限制条件下，能有多少种不同的方案来点亮这些灯泡。此外，他还想找到一种连接所有灯泡的方式，使得点亮灯泡的方案数量最大化。

给定整数 m 与 d，您的目标是协助小青鱼来确定使用所有 m 根电线连接 n 个灯泡的最佳方式，最大化点亮灯泡的方案数。请注意，您需要自行确定 n 的值。

Input

有多组测试数据。第一行输入一个整数 T（$1 \le T \le 200$）表示测试数据组数，对于每组测试数据：

第一行输入两个整数 m 与 d（$2 \le m \le 20$，$2 \le d \le m$）。

Output

对于每组测试数据：

第一行输出一个整数 w（$1 \le w \le 2^{m+1}$），表示最多可以有多少种方法点亮灯泡。

第二行输出一个整数 n（$1 \le n \le m+1$），表示小青鱼需要使用的灯泡数量。

接下来输出 m 行，其中第 i 行输出两个由单个空格分隔的整数 u_i 和 v_i（$1 \le u_i, v_i \le n, u_i \ne v_i$），表示一条连接第 u_i 个灯泡与第 v_i 个灯泡的电线。

Example

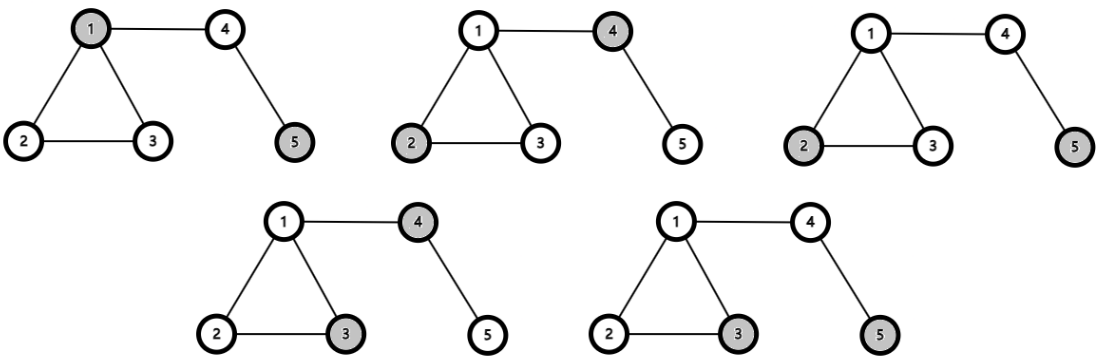
standard input	standard output
3 2 2 5 4 6 2	2 3 1 2 2 3 5 5 1 2 1 3 2 3 1 4 4 5 7 7 1 2 2 3 3 4 4 5 5 6 6 7

Note

我们用涂色的圆圈代表点亮的灯泡。

对于第一组样例数据，2 种点亮灯泡的方案如下图所示。

对于第二组样例数据，5 种点亮灯泡的方案如下图所示。



Problem D. 最大数码

令 $f(x)$ 为正整数 x 十进制表示下的最大数码。例如，$f(4523) = 5$ 以及 $f(1001) = 1$。

给定四个正整数 l_a，r_a，l_b 和 r_b 满足 $l_a \le r_a$ 且 $l_b \le r_b$，计算 $f(a+b)$ 的最大值，其中 $l_a \le a \le r_a$ 且 $l_b \le b \le r_b$。

Input

有多组测试数据。第一行输入一个整数 T（$1 \le T \le 10^3$）表示测试数据组数，对于每组测试数据：

第一行输入四个整数 l_a，r_a，l_b 和 r_b（$1 \le l_a \le r_a \le 10^9$，$1 \le l_b \le r_b \le 10^9$）。

Output

每组数据输出一行一个整数表示 $f(a+b)$ 的最大值。

Example

standard input	standard output
2 178 182 83 85 2 5 3 6	7 9

Note

对于第一组样例数据，答案是 $f(182+85) = f(267) = 7$。

对于第二组样例数据，答案是 $f(4+5) = f(9) = 9$。

Problem E. 我只想要...多一个...

在编写完论文《Sandpile Prediction on Structured Undirected Graphs》后，小青鱼想要所有人都多多解决图论问题。“没有图论问题我们便活不下去，所有人都应该来做沙堆预测问题！”

二分图是一张满足如下条件的图：它的节点可以被分成两个不相交的集合 U 与 V，使得图中的每一条边都连接 U 中的一个节点与 V 中的一个节点。如果 U 与 V 中节点的数量相同，则称这张图为平衡二分图。

无向图的匹配是一个边的集合，其中任意两条边都没有共同的端点。图的最大匹配是一个包含了最多条边的匹配。图的匹配数是这张图的最大匹配所包含的边的数量。

现在，小青鱼给了您一张平衡二分图。您需要添加恰好一条边，连接 U 中的一个节点与 V 中的一个节点，使得图的匹配数增加。求方案数。

Input

有多组测试数据。第一行输入一个整数 T 表示测试数据组数，对于每组测试数据：

第一行输入两个整数 n 与 m（$1 \le n, m \le 10^5$）表示 U 与 V 的节点数量以及边的数量。

对于接下来 m 行，第 i 行输入两个整数 u_i 与 v_i（$1 \le u_i, v_i \le n$），表示第 i 条边连接 U 中第 u_i 个节点与 V 中第 v_i 个节点。图可能有重边。

保证所有数据 $(n + m)$ 之和不超过 4×10^5。

Output

每组测试数据输出一行一个整数表示答案。

Example

standard input	standard output
3 4 3 1 2 3 2 4 3 3 3 1 3 2 2 3 1 3 2 1 2 1 2	6 0 4

Note

对于第一组样例数据，原图的匹配数为 2。通过添加边 $(1, 1)$，$(1, 4)$，$(2, 1)$，$(2, 4)$，$(3, 1)$ 或 $(3, 4)$，我们可以将匹配数增加到 3。所以答案为 6。

对于第二组样例数据，原图的匹配数为 3。显然我们无法增加匹配数，因为所有的节点都已经在匹配之中，所以答案为 0。

对于第三组样例数据，原图的匹配数为 1。通过添加边 $(2, 1)$，$(2, 3)$，$(3, 1)$ 或 $(3, 3)$，我们可以将匹配数增加到 2。所以答案为 4。

Problem F. 向未来说你好

"呃...嗯，我们可以重新开始我们的友谊吗？"
"你什么意思...你是说，重新开始？"
"..."

很久很久以前，小青鱼与他最好的朋友小 $\mathcal{F}$ 一起准备了一场程序设计竞赛。他们准备一共了 n 道题目，编号为从 1 到 n 的整数。第 i 道题目（$1 \le i \le n$）有一个难度评级 a_i。

时间过得很快。距离他们举办的比赛已经过去十五个月了。小青鱼已经不再是 Informatics Olympiad 的选手了，而是转型成为了一名教练。但他们曾有过约定，要一起举办一组系列锦标赛。

而小青鱼没有忘掉它。

现在，小青鱼想要把这 n 道题目分组为若干场训练活动。为了确保题目的题面中包含的故事背景一致，小青鱼想要把这 n 道题目划分成若干个区间。一个划分题目的方案可以记为一个整数序列 $0 = r_0 < r_1 < r_2 < \cdots < r_k = n$，表示共有 k 场训练活动，其中第 i 场活动包含所有编号在 $(r_{i-1} + 1)$ 和 r_i 之间（含两端）的题目。

除此以外，小青鱼不想让某一场活动变得太不平衡。如果一场活动中包含一道困难的题目，那么这个活动就应该包含更多的题目。形式化地，如果题目 j 在第 i 个活动中（也就是说，$r_{i-1} < j \le r_i$），那么不等式 $r_i - r_{i-1} \ge a_j$ 必须成立。

小青鱼很好奇能够有多少种划分题目的方案可以满足上述所有要求，并把方案数记做 $f(a)$。这个问题对他来说很简单，所以小青鱼很轻松地就算出了答案。

在这些活动的前一天，小青鱼突然意识到这些题目对选手来说太困难了。因此，他想到了一道新的简单题目，其难度评级只有 1。他很好奇，对每个 $1 \le j \le n$，如果我们使用如下方式定义序列 $a^{(j)}$，那么 $f(a^{(j)})$ 的值是多少。

$$a_i^{(j)} = \begin{cases} 1 & i = j \\ a_i & \text{otherwise} \end{cases}$$

因为 $f(a^{(j)})$ 的值可以非常大，您只需要输出它对 $998\,244\,353$ 取模后的结果。

Input

每个测试文件仅有一组测试数据。

第一行输入一个整数 n （$1 \le n \le 2 \times 10^5$）表示题目的数量。

第二行输入 n 个整数 $a_1, a_2, \cdots, a_n$ （$1 \le a_i \le n$），其中 a_i 表示第 i 道题目的难度评级。

Output

输出一行 n 个由单个空格分隔的整数，其中第 i 个整数表示 $f(a^{(i)})$ 的值对 $998\,244\,353$ 取模后的结果。

请不要在行末输出多余空格，否则您的答案可能会被认为是错误的！

Example

standard input	standard output
5 1 3 2 1 2	3 6 3 3 6

Note

在样例数据中，对于 $j=1$，我们有 $a^{(j)}=[1,3,2,1,2]$。有 3 种方法将题目分配到活动中，如下所示：

- $[1]$, $[3,2,1,2]$
- $[1,3,2]$, $[1,2]$
- $[1,3,2,1,2]$

对于 $j=2$，我们有 $a^{(j)}=[1,1,2,1,2]$。有 6 种方法将题目分配到活动中，如下所示：

- $[1]$, $[1]$, $[2,1,2]$
- $[1]$, $[1,2]$, $[1,2]$
- $[1]$, $[1,2,1,2]$
- $[1,1]$, $[2,1,2]$
- $[1,1,2]$, $[1,2]$
- $[1,1,2,1,2]$

对于 $j=3$ 和 $j=4$，所有的方案与 $j=1$ 时的方案相同。
对于 $j=5$，我们有 $a^{(j)}=[1,3,2,1,1]$。有 6 种方法将题目分配到活动中，如下所示：

- $[1]$, $[3,2,1]$, $[1]$
- $[1]$, $[3,2,1,1]$
- $[1,3,2]$, $[1]$, $[1]$
- $[1,3,2]$, $[1,1]$
- $[1,3,2,1]$, $[1]$
- $[1,3,2,1,1]$

Problem G. 来自知识的礼物

在学习完由理查德·彭和桑托什·文帕拉编写的论文《Solving Sparse Linear Systems Faster than Matrix Multiplication》后，小青鱼变得痴迷于一切稀疏的东西，比如稀疏矩阵。在这里，一个稀疏矩阵指零项的数量远高于非零项的矩阵。现在，小青鱼想到了一个有关二进制稀疏矩阵的问题，希望您能着手解决。

给定一个 r 行 c 列的二进制矩阵（一个仅包含 0 和 1 的矩阵），对于每一行您可以选择是否进行反转。求选择一些行进行反转的方案数（允许不选择任何行），使得每一列至多有一个 1。称两种方案是不同的，若有一行在其中一种方案里被选择，而在另一种方案里没有被选择。

本题中，反转一行的含义是这样的：设第 i 行从第一列到最后一列的元素依次为 $b_{i,1}, b_{i,2}, \cdots, b_{i,c}$。如果您反转了第 i 行，则它将变为 $b_{i,c}, b_{i,c-1}, \cdots, b_{i,1}$。

Input

有多组测试数据。第一行输入一个整数 T 表示测试数据组数，对于每组测试数据：

第一行输入两个整数 r 和 c（$1 \le r, c \le 10^6$，$1 \le r \times c \le 10^6$）表示矩阵的行数和列数。

对于接下来 r 行，第 i 行输入一个字符串 $b_{i,1}b_{i,2}\cdots b_{i,c}$（$b_{i,j} \in \{0, 1\}$），其中 $b_{i,j}$ 表示矩阵第 i 行第 j 列的元素。

保证所有数据 $r \times c$ 之和不超过 10^6。

Output

每组数据输出一行一个整数表示方案数。由于答案可能很大，请将答案对 $(10^9 + 7)$ 取模后输出。

Example

standard input	standard output
3 3 5 01100 10001 00010 2 1 1 1 2 3 001 001	4 0 2

Note

对于第一组样例数据，选择的行的集合可以是空集，$\{1, 3\}$，$\{2\}$ 或 $\{1, 2, 3\}$。所以答案是 4。

Problem H. 基本子串结构

在编写完论文《Faster Algorithms for Internal Dictionary Queries》后，小青鱼与奇异强子决定编写如下的题目。

令 $\text{lcp}(s,t)$ 表示字符串 $s = s_1s_2\dots s_n$ 与 $t = t_1t_2\dots t_m$ 的最长公共前缀，也就是最大的整数 k 满足 $0 \le k \le \min(n,m)$ 且 $s_1s_2\dots s_k$ 等于 $t_1t_2\dots t_k$。

小青鱼给了您一个非空的字符串 $s = s_1s_2\dots s_n$。令 $f(s) = \sum\limits_{i=1}^{n} \text{lcp}(s, \text{suf}(s,i))$，其中 $\text{suf}(s,i)$ 表示 s 从 s_i 开始的后缀（即 $\text{suf}(s,i) = s_is_{i+1}\dots s_n$）。请注意在本题中，字母表中包含了 n 种字母，而不是仅有 26 种。

对每个 $i = 1,2,\cdots,n$，您需要回答如下询问：如果您必须将 s_i 修改为另一个不同的字符 c（$c \ne s_i$），请选择最优的字符 c 并计算 $f(s^{(i)})$ 的最大值，其中 $s^{(i)} = s_1\dots s_{i-1}cs_{i+1}\dots s_n$。

Input

有多组测试数据。第一行输入一个整数 T 表示测试数据组数，对于每组测试数据：

第一行输入一个整数 n（$2 \le n \le 2\times 10^5$）表示字符串的长度。

第二行输入 n 个整数 $s_1, s_2, \dots, s_n$（$1 \le s_i \le n$），其中 s_i 表示字符串的第 i 个字符是字母表中第 s_i 个字母。

保证所有数据 n 之和不超过 2×10^5。

Output

令 $m(i)$ 表示 $f(s^{(i)})$ 的最大值。为了减少输出的大小，对于每组测试数据输出一行一个整数表示 $\sum\limits_{i=1}^{n}(m(i) \oplus i)$，其中 $\oplus$ 是按位异或运算符。

Example

standard input	standard output
2 4 2 1 1 2 12 1 1 4 5 1 4 1 9 1 9 8 10	15 217

Note

对于第一组样例数据，我们首先计算 $m(1)$ 的值。

- 如果将 s_1 修改为 1，那么 $f(s^{(1)}) = 4+2+1+0 = 7$。
- 如果将 s_1 修改为 3 或 4，那么 $f(s^{(1)}) = 4+0+0+0 = 4$。

因此 $m(1) = 7$。

类似地，$m(2) = 6$，$m(3) = 6$ 以及 $m(4) = 4$。所以答案为 $(7\oplus 1)+(6\oplus 2)+(6\oplus 3)+(4\oplus 4) = 15$。

Problem I. 奇怪的排序

我们给出了一个极其简单的排序算法。它看上去显然是错的，但我们证明了它事实上是对的。[a]

[a] Stanley P. Y. Fung. Is this the simplest (and most surprising) sorting algorithm ever? arXiv:2110.01111

在学习了 2021 国际大学生程序设计竞赛亚洲区域赛南京站的《Paimon Sorting》一题中奇怪的排序算法后，小青鱼想到了如下的一个问题。

给定序列 $a_1, a_2, \cdots, a_n$ 表示一个 n 的排列，您需要将该排列按升序排序，为此可以执行以下操作至多 $\lfloor \frac{n}{2} \rfloor$ 次：选择两个下标 l 和 r 满足 $1 \le l < r \le n$ 以及 $a_l > a_r$，将 $a_l, a_{l+1}, \cdots, a_r$ 按升序进行排序。

请回忆：一个 n 的排列是一个长度为 n 的序列，每个从 1 到 n（含两端）的整数在其中都恰好出现一次。另外，$\lfloor x \rfloor$ 表示小于等于 x 的最大的整数。

Input

有多组测试数据。第一行输入一个整数 T 表示测试数据组数，对于每组测试数据：

第一行输入一个整数 n（$1 \le n \le 100$）表示排列的长度。

第二行输入 n 个不同的整数 $a_1, a_2, \cdots, a_n$（$1 \le a_i \le n$）表示给定的排列。

保证所有数据 n 之和不超过 10^4。

Output

对于每组数据，首先输出一行一个整数 k（$0 \le k \le \lfloor \frac{n}{2} \rfloor$）表示您执行的操作次数。接下来输出 k 行，其中第 i 行输出两个由单个空格分隔的整数 l_i 和 r_i，表示您为第 i 次操作选择的两个下标。

可以证明答案总是存在。如果有多种合法答案，您可以输出任意一种。

Example

standard input	standard output
3 6 2 3 4 6 5 1 5 1 2 3 4 5 3 2 3 1	2 3 6 1 3 0 1 1 3

Note

对于第一组样例数据，在第 1 次操作后排列变为 $\{2, 3, 1, 4, 5, 6\}$，在第 2 次操作后排列变为 $\{1, 2, 3, 4, 5, 6\}$，此时排列是升序的。

Problem J. 计算智能

给定两条二维笛卡尔平面上的线段，您需要从每条线段上等概率随机选择一个点，并计算两点之间欧氏距离的期望值。

Input

有多组测试数据。第一行输入一个整数 T（$1 \le T \le 10^5$）表示测试数据组数，对于每组测试数据：

第一行输入四个整数 x_1，y_1，x_2 和 y_2（$-10^3 \le x_1, y_1, x_2, y_2 \le 10^3$）表示第一条线段的两个端点是 (x_1, y_1) 和 (x_2, y_2)。

第二行输入四个整数 x_3，y_3，x_4 和 y_4（$-10^3 \le x_3, y_3, x_4, y_4 \le 10^3$）表示第二条线段的两个端点是 (x_3, y_3) 和 (x_4, y_4)。

保证两条线段的长度均为正数。

Output

每组数据输出一行一个数，表示两个随机选择的点之间距离的期望值。

如果相对误差或绝对误差不超过 10^{-9}，您的答案将被接受。具体来说，设您的答案为 a，裁判的答案为 b，当且仅当 $\frac{|a-b|}{\max(1,|b|)} \le 10^{-9}$ 时，您的答案将被接受。

Example

standard input	standard output
3 0 0 1 0 0 0 1 0 0 0 1 0 0 0 0 1 0 0 1 0 0 1 1 1	0.333333333333333333 0.765195716464212691 1.076635732895178009

Note

感谢“计算智能”，我们可知：

对于第一组样例数据，距离的期望值为

$$\int_0^1 \int_0^1 |x_0 - x_1| \, \mathrm{d}x_0 \, \mathrm{d}x_1 = \frac{1}{3} \approx 0.333333333333333333;$$

对于第二组样例数据，距离的期望值为

$$\int_0^1 \int_0^1 \sqrt{x^2 + y^2} \, \mathrm{d}x \, \mathrm{d}y = \frac{\sqrt{2} + \ln(1 + \sqrt{2})}{3} \approx 0.765195716464212691;$$

对于第三组样例数据，距离的期望值为

$$\int_0^1 \int_0^1 \sqrt{(x_0 - x_1)^2 + 1} \, \mathrm{d}x_0 \, \mathrm{d}x_1 = \frac{2 - \sqrt{2} + 3\ln(1 + \sqrt{2})}{3} \approx 1.076635732895178009.$$

Problem K. 彩虹子数组

《彩虹数列》是一款紧张刺激的运气类与拍卖类桌游。玩家可以赌运气抽取更多卡牌，也可以用货币购买其它卡牌，目的是用每种颜色的卡牌构成尽量长的数字序列。接下来我们考虑一道和游戏相关的问题。

Instagram 用户 *@freethemeeple* 拍摄的照片

给定长度为 n 的序列 $a_1, a_2, \cdots, a_n$，称它的连续子数组 $a_l, a_{l+1}, a_{l+2}, \cdots, a_r$ 为彩虹子数组，若对于所有 $l \le i < r$ 都满足 $a_{i+1} - a_i = 1$。特别地，长度为 1 的子数组总是彩虹子数组。

您可以执行至多 k 次操作。每次操作您可以将序列中的一个元素增加或减少一。求完成操作后，最长彩虹子数组的长度最大是多少。

Input

有多组测试数据。第一行输入一个整数 T 表示测试数据组数，对于每组测试数据：

第一行输入两个整数 n 和 k（$1 \le n \le 5 \times 10^5$，$0 \le k \le 10^{15}$）表示序列的长度以及您最多能执行几次操作。

第二行输入 n 个整数 $a_1, a_2, \cdots, a_n$（$1 \le a_i \le 10^9$）表示序列。

保证所有数据 n 之和不超过 5×10^5。

Output

每组数据输出一行一个整数，表示至多执行 k 次操作后，最长彩虹子数组的长度最大是多少。

Example

standard input	standard output
5 7 5 7 2 5 5 4 11 7 6 0 100 3 4 5 99 100 5 6 1 1 1 1 1 5 50 100 200 300 400 500 1 100 3	4 3 5 1 1

Note

对于第一组样例数据，我们可以执行 4 次操作，并将序列变为 $\{7, 3, 4, 5, 6, 11, 7\}$。最长彩虹子数组是 $\{3, 4, 5, 6\}$，所以答案是 4。

对于第二组样例数据，我们不能执行任何操作。最长彩虹子数组是 $\{3, 4, 5\}$，所以答案是 3。

对于第三组样例数据，我们可以执行 6 次操作，并将序列变为 $\{-1, 0, 1, 2, 3\}$。整个序列都是彩虹子数组，所以答案是 5。

Problem L. 铁路环游

请注意本题不同寻常的空间限制。

《铁路环游》是一款以铁路为主题的德式桌游。在游戏中，玩家需要打出火车卡并在地图上建造铁路。建造出的铁路长度以及玩家是否能连通较远的城市决定了得分，其中需要连通的城市由抽取的车票卡决定。

BoardGameGeek 用户 *@garyjames* 拍摄的照片

考虑游戏的一维版本。有 $(n+1)$ 座城市排成一行，从左到右编号从 0 到 n。对于每个 $1 \le i \le n$，您可以在城市 $(i-1)$ 与城市 i 之间放置一条铁路以连通它们。

有 m 张车票卡用于奖励玩家连通城市的行为。第 i 张卡可以记为三个整数 l_i，r_i 和 v_i，表示如果城市 l_i 与 r_i 可以通过铁路连通（也就是说，对于所有 $l_i < j \le r_i$，城市 $(j-1)$ 和 j 之间都有一条铁路），您将获得 v_i 分。

对于每个 $1 \le k \le n$，计算您恰好放置了 k 条铁路的最大得分。如果没有获得任何奖励，您的得分为 0。

Input

有多组测试数据。第一行输入一个整数 T 表示测试数据组数，对于每组测试数据：

第一行输入两个整数 n 和 m（$1 \le n, m \le 10^4$）表示您可以放置铁路的最大数量以及用于奖励的车票卡数量。

对于接下来 m 行，第 i 行输入三个整数 l_i，r_i 和 v_i（$0 \le l_i < r_i \le n$，$1 \le v_i \le 10^9$）表示如果城市 l_i 与 r_i 可以通过铁路连通，您将获得 v_i 分。

保证所有数据 n 之和与 m 之和均不超过 10^4。

Output

每组数据输出一行 n 个由单个空格分隔的整数，其中第 i 个整数表示您恰好放置了 i 条铁路的最大得分。

请不要在行末输出多余空格，否则您的答案可能会被认为是错误的！

Example

standard input	standard output
2 4 3 0 2 3 3 4 2 0 3 1 3 1 1 3 100	2 3 5 6 0 100 100

Note

令 $(i-1, i)$ 表示一条位于城市 $(i-1)$ 和 i 之间的铁路。对于第一组样例数据：

- 如果您放置 1 条铁路，您可以放置 $(3,4)$，然后获得第二个奖励。答案是 2。
- 如果您放置 2 条铁路，您可以放置 $(0,1)$ 和 $(1,2)$，然后获得第一个奖励。答案是 3。
- 如果您放置 3 条铁路，您可以放置 $(0,1)$，$(1,2)$ 和 $(3,4)$，然后获得第一和第二个奖励。答案是 $3+2=5$。
- 如果您放置了所有 4 条铁路，可以获得所有奖励。答案是 $3+2+1=6$。

Problem M. 近似凸多边形

这是一个关于凯文的故事，他是小青鱼的朋友。

凯文是国际凸多边形大赛（ICPC）的裁判长。他为比赛准备了一道几何题。然而，由于他对计算几何不熟悉，他无法为这道题生成正确的凸多边形作为测试数据。

为此凯文很沮丧。他的好朋友小青鱼如此安慰他："虽然你生成的数据不是凸多边形，但你可以称它为近似凸多边形！"

给定一个由二维平面上的点组成的集合 S（包含至少 3 个点），其中任意两点的坐标都不相同，且任意三个点都不共线。小青鱼称一个多边形 P 为近似凸多边形，当且仅当：

- 多边形 P 是简单多边形，也就是说，多边形的顶点两两不同，且除了相邻边存在公共顶点外，不存在两条边有公共点。
- 多边形的顶点属于 S，且 S 中所有点要么在多边形内部，要么在多边形的边界上。

令 $\mathbb{U}$ 表示所有近似凸多边形构成的集合。可以证明 $\mathbb{U}$ 是一个有限集合，且不是空集合。因此，存在一个多边形 R 使得 $|R|$ 在 $\mathbb{U}$ 的所有多边形中是最小的（$|R|$ 是多边形 R 的顶点数量）。

凯文和小青鱼希望您能计算多边形 $Q \in \mathbb{U}$ 的数量，满足 $|Q| \le |R| + 1$。

Input

每个测试文件仅有一组测试数据。

第一行输入一个整数 n（$3 \le n \le 2 \times 10^3$）表示集合 S 中点的数量。

对于接下来 n 行，第 i 行输入两个整数 x_i 和 y_i（$-10^6 \le x_i, y_i \le 10^6$）表示集合 S 中的一个点 (x_i, y_i)。

保证 S 中任意两点的坐标都不相同，且任意三个点都不共线。

Output

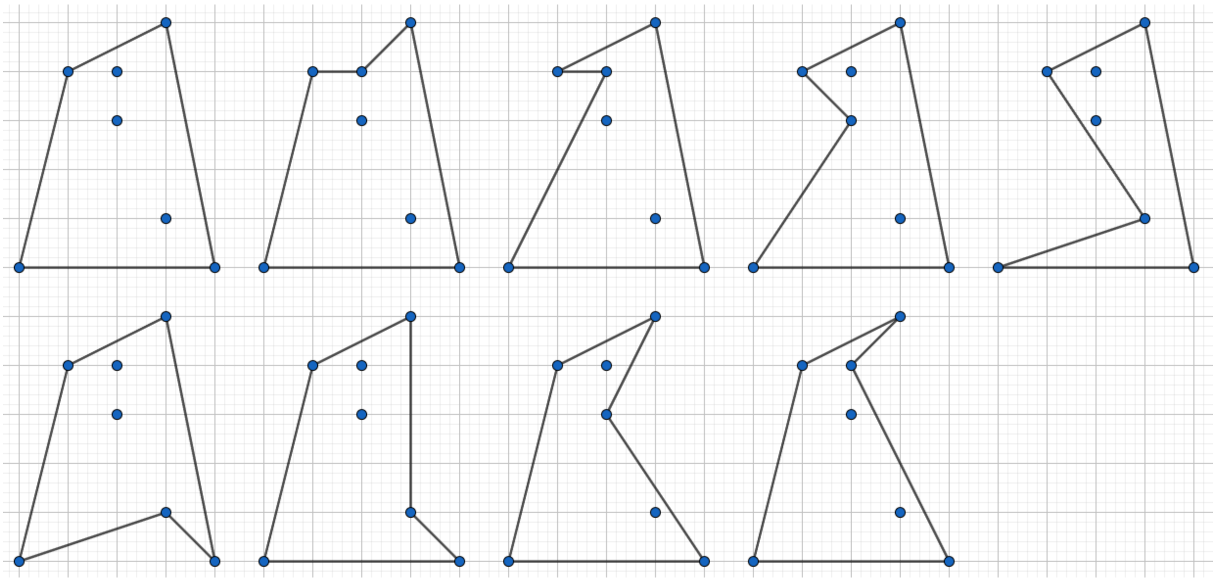
输出一行一个整数表示多边形 Q 的数量。

Examples

standard input	standard output
7 1 4 4 0 2 3 3 1 3 5 0 0 2 4	9
5 4 0 0 0 2 1 3 3 3 1	5
3 0 0 3 0 0 3	1

Note

对于第一组样例数据，$|R| = 4$。所有多边形 Q 如下所示。



对于第二组样例数据，$|R| = 3$。所有多边形 Q 如下所示。