

The 2025 CCPC
National Invitational Contest (Northeast)

The 19th Northeast Collegiate Programming Contest
(选拔赛)

Contest Session

May 04, 2025



沈阳化工大学
SHENYANG UNIVERSITY OF CHEMICAL TECHNOLOGY

Problem List

A	寂空的约数游戏
B	你有这个时间是不是能做两百道题
C	Train more !
D	NiM GaMe
E	线性探查存储器
F	Eat
G	OrzOrzOrz
H	CS , GO !
I	生命不息 , vp不止 !
J	因为要缩进
K	典题
L	G or H

This problem set should contain 12 problems on 18 numbered pages. Please inform a runner immediately if something is missing from your problem set.

Problem A. 寂空的约数游戏

输入文件: standard input
输出文件: standard output
时间限制: 1 s
内存限制: 256 MB

在沈阳化工大学的 ACM 集训队里，大家正在讨论如何在一次算法比赛中脱颖而出。为了让大家更好地理解约数枚举和贪心分配技巧，学长「寂空」提出了一个有趣的数学问题：给定一个正整数 n ，能否将它拆分成三个正约数之和，同时让这三个数的乘积最大？大家一听，都觉得既要考虑约数，又要优化乘积，思路非常具有挑战性，纷纷跃跃欲试。

给定一个正整数 n ，请找到三个正整数 x,y,z ，满足：

- 1. $n = x + y + z$
- 2. $x \mid n, y \mid n, z \mid n$ （即三者都是 n 的约数）；
- 3. 使得乘积 $x \times y \times z$ 尽可能大。

如果存在多组解，输出其中乘积最大的那个；如果不存在任何符合条件的三元组，则输出 -1 。

输入格式

第一行包含一个正整数 $T(1 \leq T \leq 10^6)$ ，表示测试用例的数量。

接下来有 T 行，每行包含一个正整数 $n(1 \leq n \leq 10^6)$ 。

输出格式

对于每个测试用例，输出一个整数——所求的最大乘积；如果无解，则输出 -1 。

样例

standard input	standard output
3	-1
1	-1
2	1
3	

提示

1 和 2 无法表示成三个数的和， $3 = 1 + 1 + 1$ 满足题意。

Problem B. 你有这个时间是不是能做两百道题

输入文件: standard input
输出文件: standard output
时间限制: 1 s
内存限制: 256 MB

好像有人很喜欢在训练的时候玩扫雷、俄罗斯方块这类小游戏啊，那既然如此，就出一道俄罗斯方块，看看你们玩明白了吗。

游戏规则如下：

首先，有一个网格：网格有 10^9 行和 w 列，从左向右数第 x 列、自底向上数第 y 行的格子用 (x, y) 表示。

在游戏刚开始（第 0 秒）时会刷新出 n 个方块，第 i 个方块刷新在 (x_i, y_i) 的位置，题目保证每个格子初始位置都不同。

每一秒，所有方块都遵循以下规则：

- 如果整个底行都布满了方块，则移除底行的所有方块。
- 对于剩余的每个方块，按照从下到上的顺序，执行以下操作：
 - 如果该方块位于最下面一行，或者它下面的单元格中有一个方块，则不做任何操作。
 - 否则，该方块向下移动一格。

给你 q 个查询，对于第 j 个查询：请回答在第 $t_j + 0.5$ 秒时第 a_j 个方块在网格中存在不存在。

输入格式

第一行输入两个整数， $n(1 \leq n \leq 2 \times 10^5)$ 和 $w(1 \leq w \leq n)$ ：格子的数量和网格的列数。

接下来 n 行：每行两个整数： $x_i, y_i(1 \leq x_i \leq w, 1 \leq y_i \leq 10^9)$ ，表示第 i 个格子的初始位置。

接下来一行一个正整数 $q(1 \leq q \leq 2 \times 10^5)$ ，询问的个数。

接下来 q 行，每行两个整数： $t_j(1 \leq t_j \leq 10^9)$ ， $a_j(1 \leq a_j \leq n)$ ，询问在第 $t_j + 0.5$ 秒时第 a_j 个方块存在不存在。

输出格式

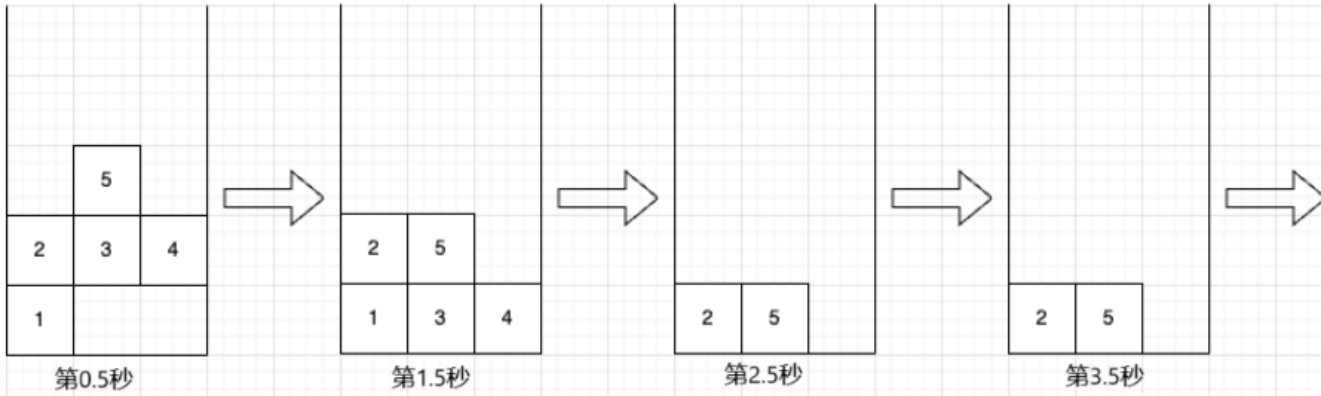
输出 q 行，询问的结果。

如果第 a_j 个方块在第 $t_j + 0.5$ 秒时存在，在一行输出 *Yes*，否则在一行输出 *No*。

样例

standard input	standard output
5 3	Yes
1 1	Yes
1 2	No
2 2	Yes
3 2	No
2 3	Yes
6	
1 1	
1 2	
2 3	
2 5	
3 4	
3 5	

提示



- 查询 1：在第 1.5 秒，块 1 存在，因此答案为 “Yes”。
- 查询 2：在第 1.5 秒，块 2 存在，因此答案为 “Yes”。
- 查询 3：块 3 在第 2 秒消失，因此它在第 2.5 秒不存在，答案为 “No”。

Problem C. Train more!

输入文件: standard input
输出文件: standard output
时间限制: 1 s
内存限制: 256 MB

在某高校的 ACM 实验室中, 有 n 名队员。起初, 他们彼此之间并不认识。由于 XCPC 的比赛都是三人团队赛, 教练决定在接下来的 m 天内, 每天早上安排两位之前互不认识的队员建立协作关系。

每个晚上, 实验室需要组织一次协作训练。参与该训练的队员应尽可能多, 但这时实验室的所有人必须满足以下规则:

- 要么这位队员不参加协作训练。
- 要么他参加协作训练, 并且至少有 k 位他已经建立协作关系的队员也参与了该日的训练。

注意: 协作关系不是传递的。如果队员 a 与 b 建立了协作关系, b 与 c 建立了协作关系, 并不意味着 a 与 c 之间也有协作关系。

教练想知道: 对于每一天, 有多少人可以参加协作训练?

输入格式

第一行包含三个整数 n 、 m 和 k ($2 \leq n \leq 2 \cdot 10^5, 1 \leq m \leq 2 \cdot 10^5, 1 \leq k < n$)—人数、天数和每个人在协作训练中应该有协作关系的人数。

接下来 m 行的 i -th ($1 \leq i \leq m$) 包含两个整数 x 和 y ($1 \leq x, y \leq n, x \neq y$), 这意味着 x 和 y 在第 i 天早上建立了协作关系。可以肯定, x 和 y 之前不是协作的关系。

输出格式

输出 m 行, 其中的 i 行 ($1 \leq i \leq m$) 表示第 i 天晚上可以参加协作训练的最大人数。

样例

standard input	standard output
4 4 2	0
2 3	0
1 2	3
1 3	3
1 4	
5 8 2	0
2 1	0
4 2	0
5 4	3
5 2	3
4 3	4
5 1	4
4 1	5
3 2	

提示

对于第一组样例，第 1、2 天没人满足条件，第 3、4 天，队员 1、2、3 都满足条件。

Problem D. NiM GaMe

输入文件: standard input
输出文件: standard output
时间限制: 1 s
内存限制: 256 MB

寂空学长在去年的 2024CCPC 全国邀请赛（东北）暨第 18 届东北地区大学生程序设计竞赛中被一道不是“NIM GAME”的“nIM gAME”惨害，痛失牌子，于是在 2025CCPC 全国邀请赛（东北）暨第 19 届东北地区大学生程序设计竞赛前的选拔赛中也出了道“NiM GaMe”来考考你们。

Mandy 对用传统的 Nim 游戏吊打 brz 已经感到索然无味，于是她决定发明一个更有趣的新游戏继续吊打 brz，规则如下：

初始有 n 堆石子，第 i 堆有 a_i 个石子。两人轮流操作，brz 先手。每次操作时，当前玩家可以选择任意一堆石子（该堆至少包含两个石子），将其拆分成两堆，且每堆至少有一个石子。拆分之后，该玩家必须将一堆中的所有石子染成白色，另一堆染成黑色。

注：染过色的石子是可以再染色的。

当所有堆都只剩下一个石子时，游戏结束。

最终，brz 的得分是场上白色石子的数量，Mandy 的得分为黑色石子的数量。两人都采取最优策略以最大化自己的得分。

游戏结束时，如果 brz 的得分严格大于 Mandy 的得分，brz 获胜，否则 Mandy 获胜。

brz 的好胜心熊熊燃起，他们一共进行了 T 轮游戏，假设两个人都足够聪明，你能在每轮游戏开始之前，提前告诉 brz 他是否能赢吗？

输入格式

第一行为一个整数 $T(1 \leq T \leq 10^4)$ ，表示游戏轮数。

对于每轮游戏，第一行为一个正整数 $n(1 \leq n \leq 10^5)$ ，第二行为 n 个整数 $a_1, a_2, \dots, a_n(1 \leq a_i \leq 10^9)$ ，每堆石子的数量。

题目保证：每轮游戏 n 的总和不超过 2×10^5

接下来有 T 行，每行包含一个正整数 $n(1 \leq n \leq 10^6)$ 。

输出格式

对于每轮游戏，输出一行。如果 brz 可以获胜，则输出“win”（不包含引号），否则输出“lose”（不包含引号）。

样例

standard input	standard output
3	lose
2	lose
1 1	lose
2	
2 2	
2	
3 3	

提示

对于这三轮游戏，无论 brz 怎么拿石子都不可能获胜。

Problem E. 线性探查存储器

输入文件: standard input
输出文件: standard output
时间限制: 1 s
内存限制: 256 MB

线性探查存储器是一种按照线性方式存储数据的结构。

它包含 $n = 2^{20}$ 个槽位，初始时每个槽位都存储 -1 ，表示为空。

其储存方式为：如果你要将数 x 储存到其中，指针会找到 $h = x \% n$ 的位置，如果位置 h 为空 ($a[h] = -1$)，就将 x 插入该位置，否则遇到冲突，指针跳到下一个位置，如果当前指针指向的位置已经是存储器的最后一个位置，指针就跳转到存储器的第一个位置，直到某一时刻指针指向的位置不为空，将 x 插入此位置。

现在你需要处理一些询问，每次询问有以下两种类型：

- 插入：将 x 插入到存储器中。
- 查询：查询当前位置 $x \% n$ 上储存的数的值。

输入格式

第一行包含一个正整数 $q(1 \leq q \leq 2 \times 10^5)$ ，询问的次数。

接下来 q 行，每行包含两个整数 $t_i(t_i \in \{1, 2\})$ 和 $x_i(0 \leq x_i \leq 10^{18})$

- 如果 $t_i = 1$ 表示插入操作
- 如果 $t_i = 2$ 表示查询操作

输出格式

对于每个查询操作，输出一行一个整数，表示查询结果，如果当前位置为空，输出 -1 。

题目保证至少存在一个查询操作。

样例

standard input	standard output
4	1048577
1 1048577	-1
1 1	
2 2097153	
2 3	

Problem F. Eat

输入文件: standard input
输出文件: standard output
时间限制: 1 s
内存限制: 256 MB

大鱼吃小鱼，小鱼吃虾米

在一条清澈的河流中，游弋着 n 条鱼，第 i 条鱼的体重为 w_i 。如果 $w_i \geq w_j$ ，鱼 i 能够吃掉鱼 j ，被吃掉的鱼会消失，吞食者的体重变为两者体重的按位异或 ($w_i \oplus w_j$)。

钓鱼的渔夫十分无聊，他想做一个参数为 x 的实验，过程如下：

1. 在河流最右端（第 n 条鱼之后）加入一条新鱼，体重为 x 。

2. 如果这条新鱼的体重不小于左侧相邻鱼的体重，则它会吞食该鱼并取代它的位置（向左移动一格）；其体重变为两条鱼体重的按位异或，重复此过程，直到左侧没有鱼或左侧鱼的体重严格大于它自身（在此过程中，其他的鱼不会自相残杀，即其他的鱼都不会被吃掉）。

3. 本次实验的得分定义为被吃掉的鱼的总数量。

渔夫给你了 q 次询问，每次给你一个整数 x ，请输出参数为 x 的实验的得分。

注意：渔夫并不希望实验真的进行（鱼都被吃了他钓什么），他只是问假设能得多少分，换句话说，询问是不持久的。

输入格式

第一行为一个正整数 $T(1 \leq T \leq 10^4)$ ，测试用例的数量。

对于每个测试用例：第一行为两个整数 $n, q(1 \leq n, q \leq 2 \times 10^5)$ ，分别表示鱼的数量和询问的数量。

下一行包含 n 个整数 $w_1, w_2, \dots, w_n(1 \leq w_i \leq 2^{30})$ ，表示鱼的重量。

接下来 q 行，一行一个整数 $x(1 \leq x \leq 2^{30})$ ，实验参数。

题目保证：所有的测试数据中， n 的总和不超过 2×10^5 ， q 的总和不超过 2×10^5 。

输出格式

每组测试用例占一行。

对于每个询问，输出一个整数：实验得分，不同询问之间以空格分隔。

样例

standard input	standard output
3	1
1 1	0 2 4 2
5	0 1 1 1 3 3 1 0 1
6	
4 4	
1 5 4 11	
8	
13	
16	
15	
10 9	
10 4 3 9 7 4 6 1 9 4	
2	
6	
5	
6	
9	
8	
6	
2	
7	

提示

- 对于第二个测试用例的第一个查询：
- 重量为 8 的鱼会被添加到最后，所以是 $w = [1, 5, 4, 11, 8]$ 。
 - 添加的鱼重量小于左边的鱼，因此它不能吃掉左边的鱼，因此在没有吃掉任何鱼后结束过程，得分 0。
- 对于第二个测试用例的第二个查询：
- 最后会添加一个重量为 13 的鱼，所以是 $w = [1, 5, 4, 11, 13]$ 。
 - 添加的鱼比它左边的鱼重量更大，因此它会吃掉它。它的重量将变为 $13 \oplus 11 = 6$ 。现在是 $w = [1, 5, 4, 6]$ 。
 - 现在，添加的鱼会吃掉它左边的鱼，它的重量变为 $6 \oplus 4 = 2$ 。现在 $w = [1, 5, 2]$ 。
 - 被添加的鱼无法再吃掉它左边的鱼了，所以它以 2 的分数结束了整个过程。

Problem G. OrzOrzOrz

输入文件: standard input
输出文件: standard output
时间限制: 1 s
内存限制: 256 MB

小弟膜拜膜拜你 Orz
小蒋有 n 名优秀的粉丝。

粉丝们根据小道消息得知小蒋将在一条直线的某一个位置出现，打算去膜拜他。为了方便，粉丝们在这条直线上建立了数轴。

第 i 名粉丝有一个侦查区间 $[l_i, r_i]$ 。如果小蒋在点 $j(l_i \leq j \leq r_i)$ 处出现，这名粉丝将立刻发现并去膜拜他。

小蒋希望膜拜他的人越多越好，但是他不能分身，因此只能选择一个位置出现。小蒋想知道自己最多能被多少个人膜拜。

输入格式

第一行一个整数 $n(1 \leq n \leq 5 \times 10^4)$ ，粉丝的个数。
接下来 n 行，每行两个整数 $l_i, r_i(1 \leq l_i \leq r_i \leq 2^{30})$ ，分别表示第 i 名粉丝的侦查区间的左右端点，两个数之间用空格隔开。

输出格式

一行一个整数，表示小蒋最多能被多少人膜拜。

样例

standard input	standard output
4 1 2 3 5 4 9 5 10	3

提示

数轴上 5 的位置会有 3 位粉丝经过，此处小蒋能被膜拜的人数最多

Problem H. CS,GO!

输入文件: standard input
输出文件: standard output
时间限制: 1 s
内存限制: 256 MB

你是 CSGO 战队的新生，为了提升自己的装备搭配水平，你找到传说中的”GO 学长” 请教武器搭配的技巧。

GO 学长告诉你，选武器不仅要看单件武器强不强，还要看主武器和副武器的配合。比如，AWP 超远射程配上 CZ75 快速近战能形成互补，AK47 强力爆头搭配 Desert Eagle 灵活爆发同样所向披靡。

学长最后给你留下了一个考验：根据一大堆武器的数据，选出一组最佳组合，让你的综合评估值最大化。能做到的话，你就能正式成为战队主力！



在 CSGO 中有 n 个主武器 (Main Weapons) 和 m 个副武器 (Secondary Weapons)。你只能各选一把主武器和副武器。每把武器都有一个综合评分 S (评分越高越好)，以及 K 个性能指标 $x[1], x[2], \dots, x[K]$ (如射程、射速、后坐力、重量……)。

为了追求武器之间的性能互补，你希望主武器和副武器在各性能指标上差异越大越好。例如，AWP + CZ75、AK47 + Desert Eagle 都是不错的组合。

对于任意一把主武器 MW 和一把副武器 SW，其总评估值计算公式为：

$$\text{Eval}(MW, SW) = S_{MW} + S_{SW} + \sum_{i=1}^K |x_i^{(MW)} - x_i^{(SW)}|.$$

现在，给定所有主武器和副武器的信息，请你选出一对武器，使得上述评估值最大，并输出该最大值。

输入格式

第一行包含一个正整数 $T(1 \leq T \leq 100)$ ，表示数据组数。

对于每组数据：

第一行包含三个正整数 $n(1 \leq n \leq 10^5), m(1 \leq m \leq 10^5), K(1 \leq K \leq 5)$ 。

接下来 n 行，每行 $K + 1$ 个整数，分别是主武器的 $S, x[1], x[2], \dots, x[K](0 \leq S \leq 10^9, |X[i]| \leq 10^9)$ 。

再接着 m 行，每行 $K + 1$ 个整数，分别是副武器的 $S, x[1], x[2], \dots, x[K](0 \leq S \leq 10^9, |X[i]| \leq 10^9)$ 。

题目保证：所有测试数据中所有 $n + m$ 的和 $\leq 3 \times 10^5$

输出格式

共 T 行，第 i 行输出第 i 组数据的最大评估值。

样例

standard input	standard output
2	543
2 2 1	2000
0 233	
0 666	
0 123	
0 456	
2 2 1	
100 0 1000 100 1000 100	
100 0	

提示

对于第一组测试数据，选第 2 把主武器和第 1 把副武器评估值最大，为 $|(0 + 666) - (0 + 123)| = 543$ 。

Problem I. 生命不息，vp 不止！

输入文件: standard input
输出文件: standard output
时间限制: 1 s
内存限制: 256 MB

有 n 位 ACMer 住一条东西走向的街道上（可以将街道看作一条线段）。

每位 ACMer 拥有一个独一无二的竞赛积分（Rating）。为了提高自己的实力、早日变成算法高手，他们经常一起 vp 一起切磋。如果两位 ACmer 想要进行切磋，他们必须从其他 ACMer 中选择一位作为裁判，并且比赛要在裁判的家中举行。出于某些原因，切磋的两位 ACmer 不能选择一个 Rating 比他们两人都高或者都低的裁判。

两位 ACmer 需要步行到裁判家中，出于贪心策略（其实就是懒），他们希望总步行距离不超过两人住址之间的距离。当然，每位 ACmer 住的位置互不相同。

如果裁判或者任意一位参赛的 ACmer 不同，则认为是两场不同的比赛。

那在这条 ACM 街上，一共可以举办多少场不同的比赛？

输入格式

第一行包含一个整数 $T(1 \leq T \leq 20)$ ，测试用例的数量。

每组测试用例包含 $n + 1$ 个整数：第一个整数为 $n(1 \leq n \leq 10^5)$ ，这条街上 ACmer 的数量，接下来 n 个整数 $a_1, a_2, \dots, a_n(1 \leq a_i \leq 10^5)$ ，表示从西到东每位 ACmer 的 Rating。

输出格式

对于每个测试用例，输出一行一个整数，表示不同比赛的数量。

样例

standard input	standard output
1 3 1 2 3	1

提示

这条街上只能举行一场比赛：第 2 位 ACMer 和第 4 位 ACMer 在第 3 位 ACMer 的家中比赛。

Problem J. 因为要缩进

输入文件:standard input

输出文件:standard output

时间限制:1 s

内存限制:256 MB

没有缩进的代码 

为什么要缩进 

因为要缩进 

在编写代码的时候，有些人习惯使用制表符（Tab 键）来控制缩进、作分隔等，而有一些人则喜欢使用空格来做这些事情。

在一些代码编辑器中，有一种将制表符转化为空格的功能。使用者只需设定好每个制表符要用多少个空格来替代，代码编辑器就会自动地对代码进行文本替代，使其代码风格与使用者接近。

这里将会给出一份 n 行的代码，你的任务是写一个程序，将代码中的所有制表符（用 `->` 表示）转化为 k 个空格，并将转化后的代码输出。

输入格式

第一行包含一个正整数 $T(1 \leq T \leq 10)$ ，表示测试数据的组数。

每组测试数据第一行包含两个正整数 $n, k(1 \leq n \leq 10, 1 \leq k \leq 8)$ ，分别表示代码的行数以及制表符应该被替换成多少个空格。

接下来 n 行，每行一个非空的字符串，表示每行代码。每个字符串长度均不超过 50，且字符串仅由小写字母 "a" 到 "z"、数字 "0" 到 "9"、特殊字符 "{", "}", "(", ")", ";", "–", ">", "*", "/", "#" 构成。

输出格式

对于每组测试数据，输出 n 行，每行一个字符串，表示每行代码转化后的结果。

样例

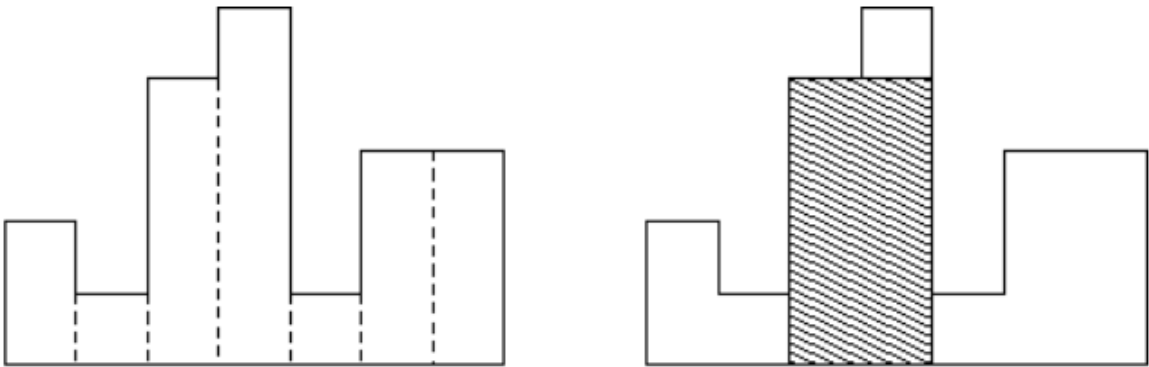
standard input	standard output
2 5 2 dfs(x){ ->if(x){ ->->dfs(x); ->} } 1 3 ->->->print(a);->print(b);	dfs(x){ if(x){ dfs(x); } } print(a); print(b);

Problem K. 典题

输入文件: standard input
输出文件: standard output
时间限制: 1 s
内存限制: 256 MB

正如标题所说，这是一道典题

直方图是由一系列沿共同基线对齐的矩形组成的多边形。这些矩形宽度相同但高度可能不同。例如，左图展示了一个由高度分别为 2、1、4、5、1、3、3（单位宽度为 1）的矩形构成的直方图：



通常直方图用于表示离散分布，例如文本中字符的出现频率。需要注意的是矩形的排列顺序（即它们的高度）非常重要。

设直方图有 n 根宽度均为 1 的矩形条，其高度依次为 $h_1, h_2, \dots, h_n$ 。

对于任意连续区间 $[i, j]$ $1 \leq i \leq j \leq n$ ，其能覆盖的矩形面积为：

$$A(i, j) = (j - i + 1) \times \min_{k=i, \dots, j} h_k.$$

则与基线对齐的最大矩形面积定义为：

$$A_{\max} = \max_{1 \leq i \leq j \leq n} [(j - i + 1) \min_{k=i}^j h_k].$$

右图便展示了该直方图对应的最大对齐矩形。

请计算直方图中与基线对齐的最大矩形面积。

输入格式

第一行为一个整数 T ($1 \leq T \leq 10$)，测试用例的数量。

接下来 T 行，每个测试用例描述一个直方图：

首先给出整数 n ($1 \leq n \leq 10^5$) 表示组成该直方图的矩形数量。

随后是 n 个整数 $h_1, h_2, \dots, h_n$ ($1 \leq h_i \leq 10^9$)，按从左到右的顺序表示各矩形的高度。每个矩形的宽度为 1。

输出格式

对于每组测试用例，输出一行一个整数，表示指定直方图中与基线对齐的最大矩形面积。

样例

standard input	standard output
2 7 2 1 4 5 1 3 3 4 1000 1000 1000 1000	8 4000

Problem L. G or H

输入文件: standard input
输出文件: standard output
时间限制: 1 s
内存限制: 256 MB

去年四省赛在 L 题的位置出了一道跟括号序列相关的题，卡住了无数队伍，今年选拔赛在同样的位置复现一下，看有没有队伍能开出来这道题呢。

众所周知，括号匹配的题一般要用栈来进行，在天梯赛没做出来括号匹配模板题后，热爱编译原理的小明手搓了一个处理括号的编译器，该编译器双栈并行，很容易出错。就在此时，编辑器接收到了一段“括号脚本”——一个只包含 “(” 和 “)” 的序列。为了保证脚本在两条独立的通道（通道 H 和通道 G）都能正确执行，必须将每个括号打上 “H” 或 “G” 的标记，要求：

- 按顺序提取所有标记为 *H* 的括号到一个序列中，该序列应该是一个合法的括号序列。
- 按顺序提取所有标记为 *G* 的括号到一个序列中，该序列应该是一个合法的括号序列。

“同一对括号” 如果标记不同，或 “不同位置的括号” 标记方式不同，都被视作不同的执行计划。小明想知道，对于给定的括号脚本，一共有多少种合法的标记方案。

输入格式

输入一行，包含一个由 “(” 和 “)” 构成的字符串 *A* ($1 \leq |A| \leq 1000$)。

输出格式

输出一行一个整数，合法的标记方案数。由于答案可能很大，所以你只需要输出答案对 998244353 取模后的结果即可。

样例

standard input	standard output
(())	6

提示

合法的标记方式有：HGGH，HGHG，GHGH，GHHG，HHHH，GGGG，共 6 种