

北京理工大学第十四届“连山科技”程序设计大赛

The 14-th BIT Campus Programming Contest

Sponsored by



现场赛

Onsite Round



题目列表

Problem List

A	两只脑斧	淡蓝
B	炼金术	白色
C	赛尔达传说	绿色
D	碟中谍	橘色
E	只有一端开口的瓶子	红色
F	风王之瞳	深蓝
G	神圣的 F2 连接着我们	紫色
H	目标是成为数论大师	黑色
I	出给 paul-lu 的数数题	灰色
J	金色传说	黄色
K	多项式求导	粉色
L	旅行的意义	浅蓝

请勿在比赛开始前翻阅试题!

Do not open before the contest has started.

2019 年 4 月 14 日

Problem A. 两只脑斧

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 128 megabytes

天天最近想要学习用口琴给箏箏吹情歌，但是他以前从来没有学过口琴，只能从最简单的两只脑斧开始学。而口琴有时吸气有时吹气的吹法让天天傻傻地分不清，所以他就只能拿着乐谱来找聪明的你了，快帮帮他！

5̣	2̣	1̣	4̣	3̣	6̣	5̣	7̣	1̣	2̣	3̣	4̣	5̣	6̣	1̣	7̣	3̣	2̣	5̣	4̣	1̣	6̣	3̣	7̣
5̣	2̣	1̣	4̣	3̣	6̣	5̣	7̣	1̣	2̣	3̣	4̣	5̣	6̣	1̣	7̣	3̣	2̣	5̣	4̣	1̣	6̣	3̣	7̣

吹 吸 吹 吸 吹 吸 吹 吸 吹 吸 吹 吸 吹 吸 吹 吸 吹 吸 吹 吸 吹 吸 吹 吸

已知天天使用的是 24 孔口琴，其音阶图如上，其中 1,2,...,7 七个阿拉伯数字表示基本的音阶，数字上方有 . 或 : 的音阶则分别表示该音阶的音调将在原音阶基础上增高一个八度或两个八度，例如音阶 3 的高八度为 3̣。相应的，数字下方有 . 或 : 的音阶也分别表示降低一个八度或两个八度的音调，例如音阶 5 的低二八度为 5̣。

Input

输入共两行，第一行输入一个正整数 n ($1 \leq n \leq 100$)，表示乐谱中有 n 个音符。

第二行输入 n 个由空格间隔开的字符串，每个字符串描述该乐谱中的一个音阶，这里使用符号 + 和 - 来分别表示音阶地高八度和低八度，例如上图的口琴乐谱的第一个音阶和最后一个音阶可以分别用 5-- 和 7+ 来表示。

Output

请输出 n 个字符描述的口琴吹法，其中第 i 个字符将描述第 i 个音阶的吹法，请用 I 表示吸气，用 E 表示吹气，用 X 表示停顿（在简谱中用 0 表示）。

Example

standard input	standard output
34 1 2 3 1 1 2 3 1 3 4 5 3 4 5 5 6 5 4 3 1 5 6 5 4 3 1 2 5- 1 0 2 5- 1 0	EIEEEIEEEIEEEIEEEIEEEIEEEIEEEXIEEX

Problem B. 炼金术

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

小白最近沉迷炼金术。通过炼金术小白可以获得很多没见过的新物质，例如 $Cu + Al \rightarrow Au + Cl$ 。

在这种炼金术中，每一种物质都可以用一个只包含小写字母的字符串来表示。同一个字符串表示相同的物质，不同的字符串表示的物质也不同。如果一种物质对应的字符串包含在另一种物质对应的字符串中（也就是它作为子串出现在另一种物质中），那么这种物质通过炼金术流程就可以得到另一种物质。例如，`ababc` 中包含了 `aba`，但 `abcba` 中不包含 `aba`。

小白认为一种物质只有它对应的字符串长度恰好为 n 时，它才是最令人愉悦的。现在小白手里已经有 m 种已知的物质。他希望找到一种新的物质，该物质不能从已有的 m 种物质中的任何一种中得到，同时也必须具有令人愉悦的特质。聪明的你能帮助他找到一种这样的物质吗？请输出该物质所对应的字符串。如果存在多种这样的物质，则你可以输出其中任意一种，题目保证至少存在一种新的物质。

Input

输入共 $m + 1$ 行，第一行输入两个正整数 n ($1 \leq n \leq 10^5$) 和 m ($1 \leq m \leq 10^4$) 由空格间隔开，表示新物质对应字符串长度 n 和小白手里已有的物质种类数 m 。

接下来 m 行，第 i 行输入一个字符串 s_i ，表示第 i 种物质对应的字符串。字符串只包含小写字母，且保证 $\sum_{i=1}^m |s_i| \leq 3 \times 10^5$ 。

Output

请输出一个长度为 n 的字符串，描述小白可以炼出的新物质。

Example

standard input	standard output
3 1 a	bit
3 2 ac ak	fun

Problem C. 赛尔达传说

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 128 megabytes

天天最近买了一台思维吃来玩赛尔达传说。

赛尔达传说需要玩家扮演近侍骑士森克从恶魔噶里噶的手里救下赛尔达公主。但是救出公主的路途异常艰险，噶里噶在去城堡的路途上设置了 n 只怪兽来阻挡森克。这 n 只怪兽分别编号从 1 到 n ，其中第 i 只怪兽血量为 d_i 攻击力为 x_i 。

已知森克的攻击力为 k ，他将按顺序轮流与这些怪兽进行若干回合制战斗。在回合制战斗中，被攻击者会损失攻击者攻击力大小的血量，当被攻击者血量掉到 0 及以下就会立即死亡。每个回合都由森克主动对怪兽发起一次攻击（森克始终先手攻击怪兽），如果怪兽尚未死亡，那么它在本回合就会立即反击森克一次，即森克被攻击后损失该只怪兽攻击力大小的血量，如此循环直到其中某一战斗方死亡为止。

森克为了确保能救出公主，提前准备了 c 份“力量炖水果”，每份“力量炖水果”都可以在瞬间吃下并使森克本回合临时增加 k 点攻击力，而且一回合内森克可以连续吃许多份“力量炖水果”。请问，森克如果想要救下赛尔达公主的话，最少需要消耗多少血量呢？



塞尔达·李达

Input

输入共 $n+1$ 行，第一行输入三个正整数 n 、 k 和 c ($1 \leq n \leq 10^5, 1 \leq k, c \leq 10^6$) 由空格间隔开，分别表示怪物数 n ，森克的攻击力 k 与他烹饪的“力量炖水果”数量 c 。

接下来 n 行，第 i 行输入两个正整数 d_i 和 x_i ($1 \leq d_i, x_i \leq 10^6$) 由空格间隔开，分别描述第 i 只怪兽的血量 d_i 与攻击力 x_i 。

Output

请输出一个非负整数，表示森克救下公主最少消耗的血量。

Example

standard input	standard output
1 5 3 21 3	3
2 3 1 3 3 4 5	0
2 10 2 51 10 20 1	31

Explanation

对于样例一，第一回合森克可以连吃三份“力量炖水果”，获得 15 点的攻击加成。此时他的攻击力达到 20 点，本回合将怪兽打到只剩 1 点血量，同时本回合他也受到了来自怪兽的 3 点伤害。

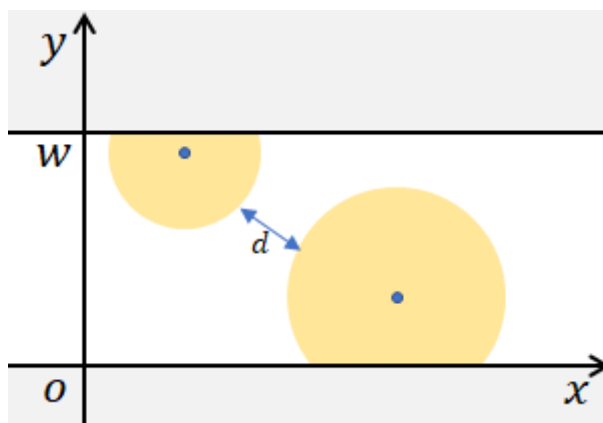
第二回合，森克的攻击加成消失，而且也没有“力量炖水果”可以吃了，所以他的攻击力回退到 5 点，但是由于怪兽仅剩 1 点血量，而且森克先手，所以他本回合可以直接将怪兽打死，不用承受本回合怪兽的伤害。因此，打完这只怪兽，森克承受的总伤害为 3 点，即他至少需要消耗 3 点血量才能救下公主。

Problem D. 碟中谍

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 128 megabytes

伊森是一个传奇的间谍人物，他所带领的“不可能任务小组”总能拯救世界于危难之中。

这次，伊森在这栋大楼里执行一项绝密的情报任务，但是面前这条幽长的走廊通道却拦住了他，走廊间安装了许多警报传感器。每个传感器都有自己的感知范围。如果人身体的某个部位在传感器感知的范围内，那么将会触动整栋大楼的中央报警装置，敌方也将会很快察觉并采取相应的措施。



为了能够在不触动报警装置的情况下悄悄通过这条走廊，伊森也许需要一些特殊的紧身装备来控制他的身材。同步协助伊森行动的小组成员班吉在收到伊森求助后，立即通过技术手段将整条走廊建模在二维平面上。如上图，整个走廊可以视为二维平面上由 $y = 0$ 与 $y = w$ 两条直线所框定的白色区域，中间有若干的报警传感器位于墙壁上或者走廊中。伊森在模型中可以视为具有一定半径的圆，位于走廊的最左侧 $x = -\infty$ ，想穿过走廊到达最右侧 $x = +\infty$ 。班吉需要尽快计算出该圆可以具有的最大半径，以便伊森可以悄悄地通过这条走廊而不触发报警装置，聪明的你快帮帮他！

Input

第一行输入一个正整数 T ($1 \leq T \leq 100$)，表示数据组数。

接下来 T 组数据，每组数据第一行输入一个正整数 w ($1 \leq w \leq 10^5$)，表示这条走廊的宽度，建模后墙壁的边缘分别为直线 $y = 0$ 与 $y = w$ 。

接下来一行输入一个非负整数 n ($0 \leq n \leq 10^3$)，表示报警传感器的数目，编号从 1 到 n 。

接下来 n 行，第 i 行输入三个整数 x_i 、 y_i 和 r_i ($-10^5 \leq x_i \leq 10^5$, $0 \leq y_i \leq w$, $1 \leq r_i \leq 10^5$) 并由空格间隔开，表示传感器建模后的坐标与感知范围。

Output

对于每组数据，请输出一个浮点数，表示能够顺利通过这条走廊而不触发报警装置的人的最大半径，注意换行。输出结果要求精确到 10^{-6} ，若精确答案为 a ，你的输出结果为 b ，只要满足 $|a - b| < 10^{-6}$ ，评测法官 Jury 就认为你的输出是正确的。

Example

standard input	standard output
3	1.500000
10	1.216991
2	0.000000
0 2 3	
12 7 4	
10	
2	
0 2 3	
8 7 4	
10	
2	
0 2 3	
4 7 4	

Problem E. 只有一端开口的瓶子

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 128 megabytes

栈就是像“只有一端开口的瓶子”一样的数据结构，大家可以理解为一个头部固定的数组，无论是取出数字还是放入数字，你都只能在数组末尾那一端（称为栈顶）修改。

非常喜欢数据结构的周老师又在给新加入 ACM 俱乐部的小萌新们传授他的人生经验。

这天，周老师得到了一个乱序的全排列。但是周老师不喜欢无序的东西，所以他总想要把这个序列弄成一个单调递增的新序列（即任意 $i < j$ 均满足 $p_i < p_j$ ）。但是他现在手里恰好只有一些简单的数据结构—— k 个栈。这些栈一开始全都为空，周老师想要只通过 3 种操作，把初始的序列变成有序的新序列：

1. 取出序列当前的第一个数字，插入到第 p 个栈的顶部：push p
2. 取出第 p 个栈的顶部数字，插入到新序列的末尾位置：pop p
3. 取出第 p 个栈的顶部数字，插入到第 q 个栈的顶部：move p q

周老师非常的睿智，他一下就想到了如果持有的栈个数大等于数字总个数（也就是 $k \geq n$ ），那么就一定可以完成这项排序工作。作为本次数据结构专题讲课的作业题，周老师想考考身为小萌新的你，至少需要多少个这样的栈才能把给定的初始序列变成有序的新序列呢？换句话说，周老师想知道 k 的最小值 $\min\{k\}$ 是多少。

Input

第一行输入一个正整数 T ($1 \leq T \leq 100$)，表示数据组数。

接下来 T 组数据，每组数据第一行输入一个正整数 n ($1 \leq n \leq 10^5$)，表示本组数据输入的全排列序列长度。

接下来一行输入 n 个正整数 p_1, p_2, \dots, p_n 由空格间隔开，描述全排列序列 P 的组成元素。输入保证 $[1, n]$ 中每个正整数在 p_1, p_2, \dots, p_n 中恰好只出现一次。注意，该序列是有顺序的，操作 1 只能从前往后取数字。

Output

对于每组数据，请输出一个正整数 k ，表示至少需要 k 个这样的栈才能把给定的初始序列变成有序的新序列，注意换行。

Example

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

Note

全排列：全排列是由 n 个数字组成的序列 p_1, p_2, \dots, p_n ($p_i \in [1, n]$)，其中数字 $1 \dots n$ 每个数字恰好只在 p_i 中恰好只出现一次。

Explanation

对于样例一，可行的方法是依次执行如下操作，即可只使用一个简单栈便处理完：

push 1 → push 1 → push 1 → pop 1 → pop 1 → pop 1

对于样例二，可行的方法是依次执行如下操作，即可只使用一个简单栈便处理完：

push 1 → pop 1 → push 1 → pop 1

对于样例三，显然一个栈无法处理，至少需要两个栈来处理：

push 1 → push 1 → move 1 2 → move 1 2 → push 2 → pop 2 → pop 2 → pop 2

Problem F. 风王之瞳

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 128 megabytes

如果莫名流泪，也许是失去了什么美好的回忆。

伪装成人类少女夏弥的龙王耶梦加得曾在暗中默默观察了这个名叫楚子航的男孩很多年。后来，她以学妹的身份正面出现在楚子航身边，一边作战，一边还讲着白烂的吐槽。她将利爪刺进他的胸膛，他也将折刀送入她的心口，他们紧紧相拥，像是最亲密的恋人。他最终来到了她生活过的小屋，仰面躺在床上，阳光温软，但眼泪却没有止住。爱唱歌的女孩被埋在花下了，连带着她的野心、和谜一样的往事。终于有一天有人问起他是不是喜欢她，他用一个问题回避了：“你们都叫她小龙女吗？”

后来，他每天都来看她曾经生活过的小屋，期待着那个虚幻的影子。终于有一天日暮，他又来到了这里，然而却被一串嬉笑声吸引——一位穿着天蓝色校服的女孩在和一个是痴呆的哥哥在花园旁的空地上画着网格玩。

突然，女孩向它招手：“楚师兄，一起来玩格点正方形的游戏嘛？”

一脸惊喜表情的楚子航却有些疑惑那是什么游戏，她真的是越来越融入人类生活了啊.....

“我来画一个 $n \times m$ 的网格图，师兄你来数一数上面有多少个正方形呀”，她调皮地眨着眼睛。

夕阳的余晖下，三人的影子显得非常的和谐。

Input

第一行输入一个正整数 T ($1 \leq T \leq 100$)，表示数据组数。

接下来 T 组数据，每组数据输入两个正整数 n 和 m ($1 \leq n, m \leq 10^5$)，由空格间隔开，表示这个网格图的长为 n ，宽为 m 。

Output

对于每组数据，请输出一个非负整数，表示这个网格图中有多个本质不同的格点正方形，注意换行。

Example

standard input	standard output
2	1
1 1	6
2 2	

Note

格点正方形：在 $n \times m$ 的网格图中，四个顶点都在格点上的正方形叫做格点正方形。

本质不同：若两个格点正方形它们至少有一个顶点不重合在同一个格点上，则将它们称作本质不同的格点正方形。

Problem G. 神圣的 F2 连接着我们

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 256 megabytes

小白非常喜欢玩“县际争霸”这款游戏，虽然他的技术并不容乐观。“县际争霸”的地图共有两个县，每个县里各有 n 个据点。同一个县之间的据点是互不连通的，两个县之间的据点也是互不连通的。小白的 p 个战斗单位在第一个县的第 x_1, x_2, \dots, x_p 个据点中，而对手的 q 个建筑单位在第二个县第 y_1, y_2, \dots, y_q 个据点中。

为了发起进攻，小白建造了很多的“折跃棱镜”。折跃棱镜可以帮助小白的单位在两个县之间移动，而且可以有多个单位同时通过折跃棱镜。具体地说，一个折跃棱镜包含有 5 个参数信息 a 、 b 、 c 、 d 和 w ，代表位于第一个县任意第 x ($a \leq x \leq b$) 个据点的战斗单位可以花费 w 单位时间到达第二个县的任意第 y ($c \leq y \leq d$) 个据点。折跃棱镜的通道是双向的，所以位于第二个县任意第 y ($c \leq y \leq d$) 个据点的战斗单位也可以花费 w 单位时间到达第一个县的任意第 x ($a \leq x \leq b$) 个据点。

如果一个小白战斗单位到达了一个有敌方建筑单位的据点，那么这个战斗单位就会即刻投入战斗。当小白所有的战斗单位都投入了战斗之后，对手会感觉到压力太大而主动投降。小白想尽快结束这场战斗，请聪明的你帮他算一算，如果采用最优的调度策略，对手最早将在什么时刻投降（假设当前局面是零时刻）。如果存在战斗单位始终无法投入战斗，则请输出 `boring game`。

Input

输入共 $m + 3$ 行，第一行输入四个正整数 n 、 m 、 p 和 q ($1 \leq n, m, p, q \leq 10^5$) 由空格间隔开，分别表示每个县的据点数量、折跃棱镜数量、小白的战斗单位数量和对手的建筑单位数量。

接下来的 m 行中，第 i 行输入五个正整数 a_i 、 b_i 、 c_i 、 d_i 和 w ($1 \leq a_i \leq b_i \leq n, 1 \leq c_i \leq d_i \leq n, 1 \leq w_i \leq 10^9$) 由空格间隔开，表示第 i 个折跃棱镜的参数，具体含义见题目描述。

接下来一行输入 p 个正整数 x_1, x_2, \dots, x_p ($1 \leq x_i \leq n$) 由空格间隔开，分别表示小白的战斗单位所在的据点位置。

最后一行输入 q 个正整数 y_1, y_2, \dots, y_q ($1 \leq y_i \leq n$) 由空格间隔开，分别表示对手的建筑单位所在的据点。

Output

如果对手最终会主动投降，则请输出一个非负整数，表示在小白的最优调度策略下对手最早的投降时间；如果存在战斗单位始终无法投入战斗，则请输出 `boring game`。

Example

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

Note

样例中，一种最优调度策略是：第一个战斗单位从第一县的 2 号据点折跃到第二县的 4 号据点即可投入战斗，这将花费 2 单位时间。同时第二个战斗单位从第一县的 3 号据点折跃到第二县的 3 号据点再折跃到第一县的 2 号据点最后折跃到第二县的 4 号据点也可投入战斗，这将花费 4 单位时间。

Problem H. 目标是成为数论大师

Input file: standard input
Output file: standard output
Time limit: 1 seconds
Memory limit: 128 megabytes

小白非常不擅长数论，在强老师的数论课上，小白听着听着又睡着了（毕竟梦里什么都有）。小白梦见自己成为了数论大师，只用了三分钟就熟练地写出了一道拉格朗日反演套多项式逆元和快速数论变换的好题。但是当他醒来一看，他还是对数论的题目一窍不通。现在就有一道简单题摆在小白的面前，聪明的你能帮帮他吗？

定义经过函数一次映射就能回到自身的点为函数的不动点 (Fix Point)。换句话说，给定函数 $f(x)$ ，它的不动点就是所有使得 $f(x) = x$ 成立的 x 的集合。如函数 $f(x) = x$ 有无穷多个不动点，而函数 $f(x) = 2x - 1$ 则只有 $x = 1$ 这一个不动点。请你找出函数 $f(x) = \sqrt{ax} + b$ 的所有不动点。

Input

第一行输入一个正整数 T ($1 \leq T \leq 100$)，表示数据组数。

接下来 T 组数据，每组数据输入两个整数 a 和 b ($-10^3 \leq a, b \leq 10^3$)，描述本组输入的函数 $f(x) = \sqrt{ax} + b$ ，保证该函数至少存在一个不动点，且所有的不动点均为整数。

Output

对于每组数据，第一行请输出一个正整数 k ，表示函数 $f(x) = \sqrt{ax} + b$ 的不动点个数。第二行请从小到大依次输出 k 个整数 x_1, x_2, \dots, x_k ($x_1 < x_2 < \dots < x_k$) 由空格间隔开，描述函数 $f(x)$ 的不动点集合，注意换行。

Example

standard input	standard output
2	2
1 0	0 1
0 1	1
	1

Note

对于样例一，函数 $f(x) = \sqrt{x}$ 存在两个不动点 $x = 0$ 与 $x = 1$ 。

对于样例二，函数 $f(x) = 1$ 只存在 1 个不动点 $x = 1$ 。

Problem I. 出给 paul-lu 的数数题

Input file: standard input
Output file: standard output
Time limit: 6 seconds
Memory limit: 128 megabytes

中国科学院大学的著名 JB 选手 bibibibi，想当年还在北理工打算法竞赛时候就是一个全才，号称北京全能王，所有的算法题目里就没有他不能嘴巴 AC 的。这天他听说 Paul(lu) Lu(pao) 学长挺会数数的，于是决定出一道数数的题目来考一考他：

已知在一个 $n \times n$ 的棋盘里，每个格子都可以填上一个范围为 $[1, k]$ 的正整数。定义棋盘中的某个格点是 bi 点 当且仅当满足：

- 该格点的整数值严格大于本行其他所有格点的整数值。
- 该格点的整数值严格大于本列其他所有格点的整数值。

设 B_i 为棋盘中恰好存在 i 个 bi 点 的方案数，请你计算 $\sum_{i=0}^{n^2} (i^2 \cdot B_i)$ ，答案可能很大，请取模 998244353。

Input

第一行输入一个正整数 T ($1 \leq T \leq 20$)，表示数据组数。

接下来 T 组数据，每组数据输入两个正整数 n 和 k ($1 \leq n, k \leq 200$)，由空格间隔开，分别表示棋盘的大小和每个格点内可填数的范围上限。

Output

对于每组数据，请输出一行，表示计算结果取模 998244353 的答案，注意换行。

Example

standard input	standard output
3	2
1 2	12
2 2	216
3 2	

Problem J. 金色传说

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 128 megabytes

龙老师获得了一个不寻常的计算器。

这个道具的品质是金色传说，它除了在按 666 时可以发出声音以外，还可以输入一个算式并计算它的结果。金色传说计算器的显示屏上只有 n 个可以显示的位置，而且只有 $0\cdots 9$ 十个数字键和两个运算符键 $+$ 、 $-$ 共 12 个按键可以按，连 $=$ 按键都没有，所以必须输入一个 n 位的合法算式，它才会自动计算出该算式的结果。

我们定义合法的算式为一个长度恰好为 n 的字符串，其中字符串只包含数字 $0\cdots 9$ 和运算符 $+$ 与 $-$ ，同时不允许算式第一个或最后一个位置出现运算符，也不允许同时有两个运算符相邻。算式的计算结果是对该字符串模拟十进制加减法运算得到的结果，只包含数字的算式其计算结果等于它本身。如果金色传说计算器的输入不是一个合法的算式，那么它就不能得到任何计算结果。例如，在 $n = 6$ 时， $1+2+3$ 、 $+12345$ 、 $23+45+$ 和 $123+-4$ 都不是合法的算式，而 123456 和 $0+0+00$ 都是合法的算式。注意，前导零是允许在算式中出现的。

龙老师一个个地数出了计算器的显示位数 n 。他想知道，如果把所有的合法的算式全都输入一次，金色传说计算器输出的所有计算结果之和是多少？计算结果的总和可能很大，请输出它取模 998244353 之后的答案。

Input

第一行输入一个正整数 T ($1 \leq T \leq 500$)，表示数据组数。
接下来 T 组数据，每组数据输入一个正整数 n ($1 \leq n \leq 5 \times 10^5$)，表示金色传说计算器的显示位。

Output

对于每组数据，请输出一个非负整数，表示所有计算结果之和取模 998244353 之后的答案，注意换行。

Example

standard input	standard output
5	45
1	4950
2	500400
3	50103000
4	19816235
5	

Explanation

当 $n = 1$ 时，共有 $0, 1, \cdots, 9$ 共 10 个合法算式，它们的结果之和是 45。

当 $n = 2$ 时，共有 $00, \dots, 99$ 共 100 个合法算式，它们的结果之和是 4950。

当 $n = 3$ 时， $000, \dots, 999$ 是合法的算式，当然与形如 $A+B$ 和 $A-B$ 的算式也都是合法算式，它们的结果之和是 500400。

Problem K. 多项式求导

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 128 megabytes

一元多项式形如 $a_nx^n + a_{n-1}x^{n-1} + \cdots + a_2x^2 + a_1x + a_0$ ，它是代数学研究的基本对象之一。求导是数学计算中一种常用的计算方法，它的表示当自变量的增量趋于零时因变量的增量与自变量的增量之商的极限。

早在高中学习导数的时候，天天就觉得给一元多次多项式求导是个简单到浪费脑细胞的工作。于是他在课间使用 C 语言写了一个给多项式求导的小程序。时隔这么多年，他早已找不到当年写的这段小程序代码了，希望聪明的你可以帮他重新完成这段给一元多项式求导的程序代码。

Input

输入共两行，第一行输入两个正整数 n 和 k ($1 \leq n, k \leq 100$)，分别表示输入的多项式次数 n ，和程序需要对该多项式进行求导运算的阶数 k 。

第二行输入 $n+1$ 个非负整数 a_n, \cdots, a_1, a_0 由空格间隔开,其中第 $n-i+1$ 个整数为 a_i ($0 \leq a_i \leq 100$)，描述输入多项式第 i 次项 x^i 的系数。

Output

请输出一行，包含 $n+1$ 个非负整数 a_n, \cdots, a_1, a_0 由空格间隔开，其中第 $n-i+1$ 个整数为 a_i ，描述输入多项式求导后第 i 次项 x^i 的系数。由于系数可能很大，对于每个 a_i 你只需要输出它取模 2019 之后的答案。

Example

standard input	standard output
5 1 1 2 3 4 5 6	0 5 8 9 8 5
5 2 1 2 3 4 5 6	0 0 20 24 18 8

Explanation

样例一和样例二的多项式为 $f(x) = x^5 + 2x^4 + 3x^3 + 4x^2 + 5x^1 + 6$ ，它的一阶导数为 $f'(x) = 5x^4 + 8x^3 + 9x^2 + 8x^1 + 5$ ，它的二阶导数为 $f''(x) = 20x^3 + 24x^2 + 18x + 8$ 。

Problem L. 旅行的意义

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 128 megabytes

为什么有人永远渴望旅行，或许就因为，巧合和温暖会在下一秒蜂拥而至吧。

一直想和箬箬去旅游的天天决定在即将到来的五一假期中安排一场环游世界的旅行。为此，他已经提前查阅了很多资料，并准备画一张旅游路线图。天天先将所有可能会去的 n 个旅游城市依次编号标记为 $1, 2, \dots, n$ 。如果从城市 A 到城市 B 有一条直达的铁路线路，那么他就会在图上画上一条从 A 向 B 的有向线段。因为天天不喜欢把时间浪费在往返的乘车上，因此他设计的旅游地图路线是一个有向无环图。

天天和箬箬身在 1 号城市，他们每到达一个旅游城市都会先花一天的时间游玩当地的旅游景点。接下来他们也没有明确的目的地，所以第二天他们会随机地选择该城市的一条直达线路，花费一天的时间通往下一个旅游城市。当然，如果这个城市的旅游景点太好玩的话，他们可能会选择再逗留一天。但是由于假期有限，他们在当前的旅游城市最多只能呆 2 天。例如，当天天和箬箬在城市 C 时，若城市 C 有 2 条直达线路分别通往城市 A 和城市 B ，则在第一天的游玩过后，第二天他们有 $\frac{1}{3}$ 的可能会选择继续逗留在城市 C 多游玩一天，但是第三天他们一定不会再逗留在城市 C 了；同时他们有 $\frac{1}{3}$ 可能会选择立即搭乘直达城市 A 的高铁；他们也有 $\frac{1}{3}$ 的可能会选择立即搭乘直达城市 B 的高铁。

当天天和箬箬把所有的旅游城市都游玩过后，他们也就只能结束这段难忘的五一旅行假期了。现在请聪明的你帮天天提前计算一下，他们本次旅行时间的期望是多少呢？

容易证明他们旅行时间的期望为 $\frac{P}{Q}$ 的形式，其中 P 和 Q 互质，且 $Q \not\equiv 0 \pmod{998244353}$ 。因此答案请以 $P \cdot Q^{-1} \pmod{998244353}$ 的形式输出，其中 Q^{-1} 表示 Q 在取模 998244353 下的逆元。

Input

第一行输入一个正整数 T ($1 \leq T \leq 10$)，表示数据组数。

接下来 T 组数据，每组数据第一行输入两个非负整数 n ($1 \leq n \leq 10^5$) 和 m ($0 \leq m \leq 10^5$)，分别表示天天和箬箬可能旅行的城市数量 n 和它们之间的直达线路数量 m 。

接下来 m 行，每行输入两个正整数 u 和 v ($1 \leq u, v \leq n$)，表示从城市 u 到 v 有一条单向直达线路，保证两个旅游城市之间最多只有 1 条直达线路。

Output

对于每组数据，请输出一个非负整数，表示天天和箬箬旅行时间的期望，注意换行。

Example

standard input	standard output
2	2
1 0	499122181
2 1	
1 2	

Explanation

样例一只有一个旅游城市。首先，天天和箬箬会在该城市游玩一天，第二天只剩下一个选择——留下来接着玩一天，再之后他们就只能结束旅程了，所以旅游时间的期望是 2。

样例二有两个旅游城市，从城市 1 到城市 2 有一条直达的线路。天天和箬箬首先在城市 1 游玩一天，然后他们有 $\frac{1}{2}$ 的概率前往城市 2，这将花费 1 天时间乘坐高铁；当然他们也有 $\frac{1}{2}$ 的概率逗留在城市 1 多玩一天，第三天再乘坐高铁前往城市 2。因此刚到达城市 2 时，他们花费的旅行时间期望是 $1 + [\frac{1}{2} \cdot 1 + \frac{1}{2} \cdot (1 + 1)] = 2.5$ 天。接着他们会在城市 2 先游玩一天，但是接下来他们就没有其他城市可以去了，只能选择继续逗留一天然后终止旅程，容易算出本次旅程总的时间期望为 4.5 天，即 $\frac{9}{2} = 9 \cdot 2^{-1} \pmod{998244353} = 499122181$ 。