

## Problem A. 卡卡杀

Input file: standard input  
Output file: standard output  
Time limit: 2.5 seconds  
Memory limit: 1024 megabytes

卡卡最近沉迷于鹅鸭杀，并且他已经不满足于 15 人局的鹅鸭杀了，所以他委托好人大脑袋火柴人设计了一款 3000 人局的鹅鸭杀。但是由于大脑袋火柴人的需要把更多的时间和精力用于开发乐乐的杏花（当前进度：已找到游戏开发相关教程，欢迎志同道合者加盟），所以大脑袋火柴人写了一个简化版卡卡杀丢给了卡卡。



卡卡杀的地图是一颗不超过 3000 个节点的树。玩家分为四个阵营，卡卡，鹅鹅，鸭鸭和中立。注意每局游戏只有一个卡卡，若干（可以为零）鹅鹅，若干（可以为零）鸭鸭和若干（可以为零）中立。每局游戏中，每名玩家都有一个起点和一个终点，并且会沿着树上最短的路径从起点走到终点，每个回合在树上移动一条边。

每名玩家都有出刀技能，发动一次技能的瞬间可以同时对他处于同一位置的所有想击倒的目标出刀，并且第  $i$  名玩家每  $t_i$  个回合会使用一次出刀（相当于技能冷却时间到了就必然会立刻出刀），即第  $i$  名玩家出刀的回合为  $t_i, 2 \times t_i, \dots, k \times t_i, \dots$ 。

每名玩家击倒其他玩家的规则如下：

- 卡卡：卡卡想无差别击倒除自己以外的所有玩家；
- 鹅鹅：鹅鹅只想击倒可怕的卡卡，不会对其他玩家造成伤害；
- 鸭鸭：鸭鸭只想击倒鹅鹅，不会对其他玩家造成伤害；
- 中立：中立想无差别击倒除自己和卡卡以外的所有玩家，不会对卡卡造成伤害。

特殊的，如果两名玩家同时发动击倒技能互相进攻，如果一方为卡卡，卡卡会被击倒而另一方在没有被别人攻击的情况下会存活（卡卡拼刀总是失败），否则双方会被同时击倒。



顺利走到终点的玩家会立刻胜利并退场（无法再被其他玩家击倒，也不会再击倒其他玩家）。

具体地，在同一回合，依次发生以下四个事件：1) 所有还在场上的玩家移动；2) 到终点的玩家获胜并退场；3) 除了卡卡以外的所有可以出刀的玩家可以自己想出刀的目标出刀，注意出刀玩家会同时对和他处于同一位置的所有想击倒的目标出刀（是的，因为是四十米大刀所以一刀可以同时砍向所有目标）。4) 如果卡卡没死，则卡卡出刀（卡卡手速太慢了）。

给出地图和玩家角色，每名玩家的起点和终点，每名玩家的技能冷却时间  $t_i$ ，按照玩家编号从小到大的顺序输出获胜玩家名单。

## Input

第一行输入两个整数  $n$  和  $m$ ，表示节点数量和玩家数量；

接下来  $n - 1$  行，每行输入两个整数  $x$  和  $y$ ，表示  $x$  和  $y$  之间有一条边；

接下来  $m$  行，每行输入四个整数  $op$ ， $tim$ ， $s$  和  $t$ ，表示玩家角色，技能冷却时间，起点和终点；

\* 若  $op = 1$ ，表示玩家为卡卡，数据保证有且只有一个卡卡；

\* 若  $op = 2$ ，表示玩家为鹅鹅；

\* 若  $op = 3$ ，表示玩家为鸭鸭；

\* 若  $op = 4$ ，表示玩家为中立。

## Output

第一行输出一个整数  $sum$ ，表示有  $sum$  个玩家存活到最后。

接下来  $sum$  行，输出存活的玩家编号，请按玩家编号从小到大输出。

**Examples**

standard input	standard output
2 1 2 1 1 30 1 2	1 1
5 4 2 1 4 2 2 5 1 3 4 3 3 5 1 1 1 4 2 2 5 4 3 1 2 3	3 1 2 4
5 5 1 2 1 3 2 4 2 5 1 2 1 5 2 3 1 4 3 1 4 2 2 1 1 2 4 2 3 4	5 1 2 3 4 5

**Note**

$1 \leq n, m, tim \leq 3000$ ,  $op \in \{1, 2, 3, 4\}$ ,  $1 \leq s, t \leq n$ .

## Problem B. 最大数码 2

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

” 当一个故事要结束的时候，我们总会想起它的开始，就像是，一切回到了从前。”

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  $la, ra, lb$  和  $rb$  满足  $la \leq ra$  且  $lb \leq rb$ ，计算  $f(a+b)$  的最大值，其中  $la \leq a \leq ra$  且  $lb \leq b \leq rb$ 。

这是艾莉第一次踏上 ICPC 亚洲区域赛赛场时遇到的签到题。转眼间，距离他开启 ACM 生涯，已经过去了整整三年。如今的他，早已能轻松解决这类入门题目。于是，他决定给这道承载着回忆的题目加点新意。



具体地说，他会给你两个正整数  $l, r$ ，满足  $l \leq x \leq r$ ，你需要：

1. 求出  $f(x)$  的最大值。
2. 求出：在所有能够令  $f(x)$  取得最大值的  $x$  中，将  $x$  转换为不含前导零的字符串  $sx$  后，字典序最小的  $sx$  是多少。

称长度为  $x$  的字符串  $P = p_1, p_2 \cdots p_x$  的字典序小于长度为  $y$  的字符串  $Q = q_1, q_2 \cdots q_y$ ，若  $x < y$  且对于所有  $1 \leq i \leq x$  有  $p_i = q_i$ ，或者

· 存在一个整数  $1 \leq t \leq \min(x, y)$ , 对于所有  $1 \leq i < t$  有  $p_i = q_i$ , 且  $p_t < q_t$ 。

## Input

每个测试文件含有多组测试数据。

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

唯一的一行输入 2 个整数  $l, r(1 \leq l \leq r \leq 10^9)$ 。

## Output

每组数据输出一行两个整数, 分别表示  $f(x)$  的最大值和在取到  $f(x)$  的最大值时的字典序最小的  $sx$ 。

## Example

standard input	standard output
4	7 17
11 17	9 19
8 39	9 19
5 27	2 2
1 2	

## Problem C. 圆桌会议

Input file: standard input

Output file: standard output

Time limit: 2 seconds

Memory limit: 256 megabytes

在须弥教令院，学者们素来有在智慧宫的圆桌前交流学术的习惯。见此情景，纳西妲便打算每周固定举办一场「圆桌会议」，而学者们的座位安排事宜，她全权交给了流浪者来负责。

须弥的学者足有  $n$  位，他们的能力高低、研究方向各有不同，恰好可以用 1 到  $n$  的一个排列来标注每个人的能力值。圆桌会议采用环形落座，每一位学者都会和自己左右相邻的两位同伴组成一个「讨论小组」，整场会议下来，一共会诞生  $n$  个这样的小组。每个小组的讨论话题，都由三人里能力最为中庸的学者来主导，因此小组的话题编号，正是三人能力值的中位数。纳西妲不希望会议的讨论内容太过分散，便要求最终的话题种类数越少越好。这份难题如今摆在了流浪者面前，但他却没能找到很好的方案，因此他找到了你，你能帮他找到合适的安排方案吗？



### Input

每个测试文件含有多组测试数据。

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

唯一一行输入 1 个整数  $n(4 \leq n \leq 10^6)$ ，表示学者的数量。

保证单个测试文件所有数据  $n$  之和不超过  $10^6$ 。

### Output

对于每组数据，输出一行  $n$  个整数，且为 1 到  $n$  的排列，表示学者的座位安排。

若有多种可能的答案，你可以输出任意一种。

**Example**

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

## Problem D. 格兰法洛

Input file:           standard input  
 Output file:         standard output  
 Time limit:          5 seconds  
 Memory limit:       1024 megabytes

在伊比利亚的沉寂之镇——格兰法洛，黄金时代的余晖早已被灰暗的阴云掩盖。蔓延的” 溟痕” 如同有呼吸的活物，正一点点吞噬着这座小镇残存的街道。

为了开辟通往海岸的道路，小 H 委托罗德岛干员卡卡来清理小镇主干道上的溟痕。这条主干道可以看作一条数轴，上面有  $n$  个待清理的区块，按位置编号为  $1, 2, \dots, n$ 。受深海侵蚀程度不同，第  $i$  个区块的溟痕浓度为  $p_i$  ( $1 \leq p_i \leq W$ )。

工程部紧急调用了一台由伊比利亚前修道院侍从改装的清洁机器——” 小帮手”。卡卡需要操控” 小帮手” 进行  $m$  次清理操作。第  $i$  次操作可以表示为三元组  $(op_i, x_i, w_i)$ ，其中  $w_i$  为本次设定的清洁功率 ( $1 \leq w_i \leq W$ )：

- 若  $op_i = 0$ ，” 小帮手” 会从主干道左侧推进，清理所有满足  $1 \leq j \leq x_i$  且浓度  $p_j < w_i$  的区块  $j$ ；
- 若  $op_i = 1$ ，” 小帮手” 会从主干道右侧推进，清理所有满足  $x_i \leq j \leq n$  且浓度  $p_j < w_i$  的区块  $j$ 。

现已知所有区块的浓度  $p_i$  以及每次操作的方向  $op_i$  和范围  $x_i$ ，请问有多少种为机器设定功率  $(w_1, w_2, \dots, w_m)$  的方案，能够使得每个区块至少被成功清理一次？

请输出满足条件的方案数对  $10^9 + 7$  取模后的结果。两种方案不同，当且仅当存在至少一个  $i \in \{1, 2, \dots, m\}$ ，使得这两种方案设定的功率  $w_i$  不同。

### Input

本题一个测试点内包含多个测试用例。

第一行输入一个正整数  $T$  ( $1 \leq T \leq 5 \times 10^3$ ) 表示测试用例的个数。对于每个测试用例：

第一行输入三个正整数  $n, m, W$  ( $1 \leq n, W \leq 5 \times 10^3$ ,  $1 \leq m \leq 10^6$ ) 分别表示区块个数、清理次数以及溟痕浓度的上限。

第二行输入  $n$  个正整数  $p_1, p_2, \dots, p_n$  ( $1 \leq p_i \leq W$ ) 表示各个区块的溟痕浓度。

接下来  $m$  行，第  $i$  行输入两个正整数  $op_i, x_i$  ( $op_i \in \{0, 1\}$ ,  $1 \leq x_i \leq n$ ) 表示第  $i$  次清理的参数。

保证所有测试用例的  $n$  之和不超过  $5 \times 10^3$ ，所有测试用例的  $m$  之和不超过  $10^6$ 。

### Output

对于每个测试用例，输出一行一个整数，表示满足” 所有区块至少被清理一次” 的功率设定方案数，结果对  $10^9 + 7$  取模。

**Example**

standard input	standard output
2	18
3 3 3	2897
1 2 1	
1 1	
1 2	
0 3	
5 5 6	
5 1 4 2 3	
0 4	
0 2	
0 1	
1 3	
0 5	

## Problem E. 乐乐的杏花

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

乐乐想去看杏花，并一直在看杏花的路上。

乐乐一直在看杏花的路上，可惜世界太大，杏花的花期太短，在某一时刻，某个地方的杏花开了，在某一时刻，某个地方的杏花落了，乐乐深知，永远走向离她最近的杏花，也不一定能在花落前抵达，所以她采取扔骰子的方式决定走哪一条路。

但是她永远想知道离她最近的盛开的杏花离她有多远。

世界地图被乐乐归纳成了张有  $n$  个点， $n - 1$  条边的连通图，每个点都有一颗杏花树。乐乐在写完她那个复杂的动态乱序多发射处理器后，决定花费  $m$  个单位时间去寻找杏花，也就是说，在  $m$  个单位时间内，每个单位时间会发生下列三种事件之一。

- \* 某一个地方的杏花开了
- \* 某一个地方的杏花落了
- \* 乐乐移动到了一个和当前节点相邻的节点

乐乐初始时在 1 号点。

### Input

第一行输入两个整数  $n$  和  $m$ ，表示节点数量和时间；

接下来  $n - 1$  行，每行输入两个整数  $x$  和  $y$ ，表示  $x$  到  $y$  之间有一条边。

接下来  $m$  行，每行输入两个整数  $op$  和  $x$ ：

\* 若  $op = 1$ ，表示点  $x$  的杏花状态变了，若点  $x$  处之前有杏花盛开，那么现在它会凋谢，若点  $x$  处之前没有杏花盛开，那么它会盛开。

\* 若  $op = 2$ ，表示乐乐移动到了点  $x$ ，保证点  $x$  与乐乐在此之前所在的点相邻

### Output

输出  $m$  行，第  $i$  行表示在  $i$  时刻离乐乐最近的盛开杏花到乐乐的距离，如果还没有杏花盛开，输出 "2147483648"；

**Example**

standard input	standard output
3 4	2147483648
1 2	0
2 3	0
2 2	1
1 2	
1 1	
1 2	

**Note**

$1 \leq n, m \leq 10^5$ ,  $op \in \{1, 2\}$ ,  $1 \leq x \leq n$ 。

## Problem F. 复读机

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

卡卡捡到了一个神奇的复读机。

如果你给它输入整数 '1'，它会输出 'I love Heilongjiang'。如果你给它输入整数 '0'，它会输出 'I love CCPC'。如果输入的是其他整数，它会输出 'Error'。

现在给定一个整数  $x$ ，请你输出这台复读机的反应。

### Input

一行一个整数  $x$  ( $-10^9 \leq x \leq 10^9$ )。

### Output

如果  $x = 1$ ，输出 'I love Heilongjiang'。如果  $x = 0$ ，输出 'I love CCPC'。否则输出 'Error'。

### Examples

standard input	standard output
1	I love Heilongjiang
0	I love CCPC
-1	Error

## Problem G. 卡卡数

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

给定一个非负整数  $x$ ，考虑区间  $[0, 10^x - 1]$  中的所有整数。

问：把这些整数都写成十进制后，数字字符 ‘1’ 一共出现了多少次？

答案对 998244353 取模。

(提示：当没有思路的时候，通过对小范围数据打表，或许可以找到一些规律或想法)

### Input

一行一个整数  $x$  ( $0 \leq x \leq 1000000$ )。

### Output

输出一个整数，表示答案对 998244353 取模后的结果。

### Examples

standard input	standard output
2	20
0	0

### Note

当  $x = 2$  时，区间是  $[0, 99]$ 。为了写出这一百个数字，我们一共要写出 20 个数字 ‘1’，所以答案是 20。

当  $x = 0$  时，区间里只有整数 0，答案是 0。

## Problem H. 周二会员日

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

”周二会员日，第二份仅 1 元，折合单堡 6.5 元！”

柠檬的学校新开了一家汉堡店，打出了这样的宣传语。爱吃汉堡的柠檬马上就关注到了这家店，可他却漏看了”折合”两个字，把宣传语的意思完全理解错了。

宣传语原本想表达的是：汉堡不参与优惠的原价为 12 元一个，参加会员日活动时，买两个汉堡只需付”1 个原价加 1 元”，合计 13 元，折算下来平均每个汉堡 6.5 元。

但柠檬没看到”折合”，理解成了：汉堡的原价就是 6.5 元一个，第二份只需要再花 1 元，所以花 7.5 元就能买到两个汉堡。

现在已知柠檬按自己的错误理解，算出来的参加活动后的单个汉堡平均价格，请你计算出汉堡不参与优惠的原价是多少。



## Input

每个测试文件含有多组测试数据。

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

唯一一行输入 1 个实数  $x(2 < x \leq 4 \times 10^{16})$ ，表示柠檬心中参加活动后的单个汉堡价格。

保证  $x$  包含小数点，且小数点后有两位数字。

## Output

对于每组数据，输出一行表示汉堡不参与优惠的原价是多少。

你的答案必须为两位小数，与**标准答案完全相同**才视为通过。

## Example

standard input	standard output
3	12.00
3.75	446818.08
111705.27	144115188075855869.04
36028797018963968.01	

## Note

提示：由于浮点数的存储方式，它在很多时候会不可避免地出现精度问题。而在实际应用中，对于金钱的处理无法忍受任何的误差：0.01 元的误差都会导致账对不上，引发各种问题。那我们应该如何去处理呢？

## Problem I. 假想对冲

Input file: standard input

Output file: standard output

Time limit: 3.5 seconds

Memory limit: 512 megabytes

罗德岛的 PRTS 系统正在进行一项兵棋推演。

推演的战术网络是一棵包含  $n$  个节点、以 1 为根的有根树。对于树上的每一个节点  $i$  ( $2 \leq i \leq n$ ), 都有一条单向边连向其父节点  $f_i$ , 且该边具有固定的通行上限  $w_i$ 。

PRTS 系统需要对这棵树上的每一棵子树分别进行独立的战术评估。对于以节点  $x$  为根的子树, 其推演规则如下:

- 演习开始前, 你可以在该子树内除  $x$  之外的任意节点上, 部署任意数量的侦察无人机。
- 演习开始后, 无人机将进行多轮同步移动。在每一轮中, 子树内除  $x$  外的任意节点  $i$  上当前停留的所有无人机, 将同时顺着单向边  $(i, f_i)$  移动至其父节点  $f_i$ 。特别地, 移动到达节点  $x$  的无人机会脱离推演区域并被系统回收, 不再参与后续移动。
- 在任意一轮移动开始前, 若子树内存在某个节点  $i$  ( $i \neq x$ ), 满足当前停留在此处的无人机总数严格大于边  $(i, f_i)$  的通行上限  $w_i$ , 则发生拥堵, 针对该子树的推演宣告失败。

我们将一棵子树的战术权值定义为: 在保证推演全程不发生拥堵的前提下, 演习开始前能够在该子树中部署的无人机总数的最大值。特别地, 仅包含单个节点 (即叶子节点) 的子树, 其战术权值为 0。

现给定上述完整的战术网络, 请你分别求出以节点  $1, 2, \dots, n$  为根的子树的战术权值。

### Input

每组测试包含多个测试用例。第一行包含一个整数  $T$  ( $3 \leq T \leq 10^5$ ) 表示测试用例的数量。对于每个测试用例:

第一行包含一个整数  $n$  ( $2 \leq n \leq 10^6$ ) 表示节点个数。

第二行包含  $n - 1$  个整数  $f_2, f_3, \dots, f_n$  ( $1 \leq f_i < i$ ) 表示各节点的上级节点。

第三行包含  $n - 1$  个整数  $w_2, w_3, \dots, w_n$  ( $1 \leq w_i \leq 10^9$ ) 表示各边的通行上限。

保证所有测试用例的  $n$  之和不超过  $3 \times 10^6$ 。

### Output

对于每个测试用例, 输出一行  $n$  个整数表示答案, 第  $i$  个整数表示以  $i$  为根的子树的权值。

**Example**

standard input	standard output
3	1124 0 0
3	7 6 0 0 0
1 1	23 15 15 9 17 8 3 11 4 0
325 799	
5	
1 1 2 2	
3 1 2 4	
10	
1 2 3 4 5 6 7 8 9	
8 3 6 2 9 5 1 7 4	

## Problem J. 秋水横渡

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

随着武陵工业 325 期的开展，武陵地区的生态发生了巨大的变化。现如今，清波寨到武陵城的水路上按顺序分布着  $n$  座关口，编号依次为  $1, 2, \dots, n$ 。乘坐竹筏从  $i-1$  号关口航行至  $i$  号关口需要花费  $t_i$  单位的时间 ( $2 \leq i \leq n$ )。

每座关口均设有周期性的通行限制：关口会在“开启”和“关闭”两种状态之间交替切换，每次维持某一状态的时间均为  $k$  个单位时间（即一个完整的开闭周期长度为  $2k$ ）。当关口处于开启状态时，竹筏可以瞬间渡过（不花费时间）；当关口处于关闭状态时，竹筏必须在关口前等待，直到关口切换为开启状态方可通行。

作为终末地的干员，小 H 在伊冯的帮助下侵入了中枢系统。他拥有最高权限，可以在出发前，任意设定每一座关口的初始启动时间。然而，由于水路环境的复杂性与系统底层的不可控延迟，小 H 抵达起点的时机无法与系统精准同步。通俗地说：当小 H 在时刻 0 恰好到达 1 号关口前准备出发时，1 号关口刚好处于周期的哪个时刻是完全随机的。

形式化地说：

- 每个关口都有一个时钟  $S_i$  ( $S_i \in [0, 2k)$ ,  $S_i \in \mathbb{R}$ )。小 H 在到达 1 号关口前，可以任意设定所有  $S_i$  ( $1 \leq i \leq n$ ) 的值。
- 系统中存在一个不可控的全局未知偏移量  $\Delta$  ( $\Delta \in \mathbb{R}$ )，并且  $\Delta$  在  $[0, 2k)$  的连续区间内独立且均匀分布。
- 时钟随着时间的推移连续且均匀地流逝。在出发后的任意时刻  $t$  ( $t \geq 0$ ,  $t \in \mathbb{R}$ )， $i$  号关口的实际时钟值为  $T_i(t) = (S_i + \Delta + t) \bmod 2k$ 。
- 在时刻  $t$ ，若  $T_i(t) < k$ ，则该时刻  $i$  号关口开启，允许通行；若  $T_i(t) \geq k$ ，则该时刻  $i$  号关口关闭，必须等待。

现在，小 H 需要乘坐竹筏依次渡过这  $n$  座关口前往武陵城。已知他会采取最完美的策略来设定所有的  $S_i$  以最小化行程耗时。请问从时刻 0 出发开始计时，直到小 H 成功渡过最后一座 ( $n$  号) 关口，所花费总时间（包含航行时间与所有等待时间）的最小数学期望是多少？

### Input

本题一个测试点内包含多组测试用例。

第一行输入一个整数  $T$  ( $1 \leq T \leq 10^5$ ) 表示测试用例的个数。对于每组测试用例：

第一行输入两个整数  $n, k$  ( $2 \leq n \leq 2 \times 10^5$ ,  $1 \leq k \leq 10^9$ ) 分别表示关口数量和每座关口的状态切换时长。

第二行输入  $n-1$  个整数  $t_2, \dots, t_n$  ( $1 \leq t_i \leq 10^9$ ) 表示竹筏在相邻两座关口之间的通行时间。

保证所有测试用例的  $n$  之和不超过  $2 \times 10^5$ 。

## Output

对于每个测试用例，输出一行实数表示花费总时间的数学期望，如果您的答案与标准答案的相对误差或绝对误差不超过  $10^{-9}$ ，则认为您的答案是正确的。

具体来说，假设您的答案是  $a_i$ ，标准答案是  $b_i$ ，则当且仅当  $\frac{|a_i - b_i|}{\max(1, |b_i|)} \leq 10^{-9}$  时，您的答案才会被接受。

## Example

standard input	standard output
2	27.750000000
4 27	50.750000000
9 2 10	
6 51	
3 3 16 6 10	

## Problem K. 卡卡拆

Input file:            standard input  
 Output file:           standard output  
 Time limit:            8 seconds  
 Memory limit:         256 megabytes

大脑袋火柴人教卡卡怎么玩鹅鸭杀，大脑袋说：“这个游戏由 13 个玩家构成，所以你要先学会数数，这是一号，这是二号，……”。总之，卡卡花了十分钟的时间完全了解了这个游戏，当然也就学会了数数。

卡卡学会了数数之后，觉得自己很会数数，于是就出了一道数数的题目，聪明的你也一定很擅长计数吧，请你先解决这个小问题，再去玩鹅鸭杀。

对于非负整数  $s$ ，一种“卡卡拆分方案”指把  $s$  表示成若干个正整数之和，且不计顺序，并满足额外限制：对于每个整数  $i$ ，它在拆分中出现的次数只能是 0 或奇数 (1, 3, 5, …)。

例如，4 的合法卡卡拆分方案只有两种：4 与 3 + 1。

2 + 2 中 2 出现了 2 次，2 + 1 + 1 中 1 出现了 2 次，它们都不合法。注意 3 + 1 和 1 + 3 视为同一种方案。

对于正整数  $n$ ，记  $p(n)$  为  $n$  的卡卡拆分方案数，并规定  $p(0) = 1$ 。

给定  $N$ ，你需要输出  $p(0), p(1), \dots, p(N)$  对 998244353 取模后的结果。

### Input

一行一个整数  $N$  ( $0 \leq N \leq 1000000$ )。

### Output

输出一行  $N + 1$  个整数，依次为  $p(0), p(1), \dots, p(N)$  对 998244353 取模后的结果。

### Example

standard input	standard output
10	1 1 1 3 2 5 6 9 9 16 20

### Note

当 4 被拆分时，只有 4 和 3 + 1 两种方案合法，所以  $p(4) = 2$ 。像 2 + 2、2 + 1 + 1 这样的拆分都不合法，因为某个数出现了偶数次。

## Problem L. 猪猪车

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

这是一道交互题。



乐乐和小诺有一辆猪猪车，猪猪车的车牌号是猪 Axxxx。

你想要猜出 xxxx 这个数字是多少 ( $0 \leq x \leq 9$ )，每次你可以询问一个四位数字，小诺会告诉你这个数字与答案有几位相同（位置和数码都一样才算相同），请在 12 次询问以内猜出这个数字。

### Interaction Protocol

首先，你需要读入一个正整数  $T(T \leq 1500)$  表示数据组数，接下来进行  $T$  局猜数字。

要提出询问，请以 "?x" 的格式输出一行，其中  $x$  为一个可以包含前导 0 的四位数字。在清空输出缓冲区后，你需要读入一个  $0 \sim 4$  之间的数字，表示你问的值和答案有几位相同。

要猜测答案，请以 “! x” 的格式输出一行，表示你的猜测结果，其中  $x$  为一个可以包含前导 0 的四位数字。在清空输出缓冲区之后，你的程序应该马上开始进行下一局猜数字。如果当前已经是最后一局，你的程序应该立即退出。请注意，猜测不算一次询问。即你可以进行 12 次询问和 1 次猜测。

每组测试数据的答案都是预先确定的。也就是说，裁判程序并不是适应性的。

清空输出缓冲区可以使用以下方式：

- C 和 C++ 使用 `fflush(stdout)`（如果您使用 `printf`）或 `cout.flush()`（如果您使用 `cout`）。
- Java 和 Kotlin 使用 `System.out.flush()`。
- Python 使用 `stdout.flush()`。

### 交互样例

standard output	standard input
	1
? 1234	
	2
? 1314	
	4
! 1314	

## Problem M. 六个核桃

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

小核桃正在整理精品核桃礼盒，他将编号为  $1 \sim n$  的核桃排成一列，对应位置编号也为  $1 \sim n$ ，最终的摆放顺序构成一个  $1 \sim n$  的排列  $p$ ，其中  $p_i$  表示第  $i$  个位置摆放的核桃编号。

为了衡量礼盒的观赏价值，小核桃定义了单颗核桃的价值得分为：

$$\min(|p_i - i|, m)$$

其中  $p_i$  表示第  $i$  颗核桃的位置， $m$  为给定的常数。

即核桃实际位置与编号的距离，与定值  $m$  取较小值。小核桃希望所有核桃的总价值得分最大化，请你构造出满足条件的任意一个排列  $p$ 。

### Input

一行，输入两个数  $n, m$ ，保证  $1 \leq n \leq 1000000$ ， $1 \leq m \leq n$ 。

### Output

输出一行整数，表示你构造出的排列。

### Example

standard input	standard output
5 2	3 4 5 1 2