

# 2019 年华南理工大学程序设计竞赛 (春季赛)



华南理工大学  
South China University of Technology

主办 华南理工大学计算机科学与工程学院



赞助商 华为

2019 年 4 月 13 日

# Problems

Problem A. NB 群友	Page 1
Problem B. 修仙时在做什么？有没有空？可以来炼丹吗？	Page 2
Problem C. 六学家的困惑	Page 4
Problem D. 神经网络	Page 5
Problem E. 数独挑战	Page 7
Problem F. 翻牌游戏	Page 8
Problem G. 铁树开花	Page 9
Problem H. Parco_Love_GCD	Page 11
Problem I. 炒股	Page 12
Problem J. 单身狗救星	Page 13
Problem K. Parco_Love_String	Page 14
Problem L. 石头剪刀布	Page 15

题目编号	气球颜色	时间限制 (毫秒, C++)	内存限制 (MByte)	输入输出
A	橙色	4000	128	stdin/stdout
B	银灰	5000	128	stdin/stdout
C	粉色	1000	128	stdin/stdout
D	棕色	4000	128	stdin/stdout
E	紫色	1000	128	stdin/stdout
F	深绿	2000	256	stdin/stdout
G	乳白	1000	128	stdin/stdout
H	红色	2000	128	stdin/stdout
I	深蓝	1000	128	stdin/stdout
J	紫红	1000	128	stdin/stdout
K	浅绿	1000	128	stdin/stdout
L	黄色	1000	128	stdin/stdout

## 注意事项：

- 1、本次比赛支持 C/C++、Java 和 Python3，题目中给出的时间限制为 C/C++ 的时间限制，Java 的时间限制为 C/C++ 的两倍，Python3 的时间限制为 C/C++ 的三倍。
- 2、允许选手携带参考书、打印资料等纸质参考资料。禁止使用任何电子设备，包括但不限于手机、U 盘、计算器、具有通讯功能的手表或手环，禁止与除裁判及志愿者外的其他人员进行交流。若发现有作弊行为，将取消选手成绩。
- 3、对于恶意提交卡评测机的行为，一经发现将取消选手成绩。
- 4、对题目有疑问可在答疑平台中提问（只会回答关于题意的问题），电脑出现问题可举手示意。
- 5、若对于题目有任何修改，请以现场裁判的说明为准。

## Problem A. NB 群友

Input file:        standard input  
Output file:      standard output

CC 是著名的算法竞赛选手，他不仅人长得帅，而且技术了得，自然而然就有了许多粉丝。

为了能帮助粉丝们提高竞技水平，CC 建立了一个粉丝群，每天 CC 都会在粉丝群里和群友深入交流黑科技。然而，有些群友老是不努力训练，成天想着复读，比如当 CC 在群里面发了个整数 0，那紧接着就会有群友发整数 1，然后又会有群友发整数 2……这引起了 CC 的不满，于是 CC 决定踢掉一些群友。

CC 的粉丝群人数为无穷大。当 CC 发出整数 0 后，其他群友就会跟着轮流发整数  $1, 2, 3, 4, \dots$ ，依此类推。需要注意的是，每个群友都会恰好发一次整数，每个群友发的整数两两不同。CC 认为，在不考虑前导零的情况下，如果某个群友发的整数在十进制表示下的各位数字不含 0 及 1，那么这个群友就是 NB 的，否则就是不 NB 的。例如，群友 A 发的整数是 3482，该数的各位数字分别为 3、4、8、2，其中不含 0、1，因此群友 A 是 NB 的；另一方面，群友 B 发的整数 402，而该数的十位数字是 0，因此群友 B 是不 NB 的。

现在 CC 决定，踢掉所有不 NB 的群友。于是经过一番奥妙重重的踢人操作后，粉丝群里只剩下 NB 群友。然而，CC 觉得剩下的这些 NB 群友还是 too naive，因此他打算邀请一些 NB 群友参加线下训练营（来现场教做人）。具体来说，CC 会给出两个正整数  $L, R$ ，然后他会邀请发出的整数的各位数字之积在区间  $[L, R]$  内的 NB 群友。

举例来说，假如 CC 给出的区间为  $L = 50, R = 300$ ，那么发了 567 的群友会被邀请线下参赛，因为  $5 \times 6 \times 7 = 210$ ；同理，发了 255 的群友也会被邀请，因为  $2 \times 5 \times 5 = 50$ 。但是，发了 328 的群友则不会收到邀请，因为  $3 \times 2 \times 8 = 48 \notin [50, 300]$ 。

现在 CC 想知道，他究竟会邀请多少 NB 群友参加线下训练营（线下自闭）呢？

### Input

第一行是一个整数  $T$  ( $1 \leq T \leq 50$ )，表示数据组数。

接下来  $T$  行，每行两个整数  $L, R$  ( $1 \leq L \leq R \leq 2^{32} - 1$ )，表示一组询问。

### Output

输出共  $T$  行。对于每个询问，输出一行一个整数  $ans$ ，表示 CC 邀请参加线下训练营的 NB 群友人数模  $10^9 + 7$  的结果。

### Example

standard input	standard output
4	7
3 6	13
4 9	793516016
2147483648 4294967295	1
5 5	

## Problem B. 修仙时在做什么？有没有空？可以来炼丹吗？

Input file: standard input

Output file: standard output

众所周知，杨大佬很强（从名字就可以看出来），但他不仅仅满足于，还想要变得更强，所以他每天都在爆肝修仙打 cf。然而过了一段时间，杨大佬发现，他已经变得反手无力，正手不精，脚步松散，反应迟钝，肝也爆不了，仙也修不动，有时甚至会产生“1 分钟只有 59 秒”这样的错觉。为了恢复往日的雄风，杨大佬采取了一系列措施——比如在修仙的时候顺便炼些丹药拿来补肝，于是很快，杨大佬又变得和过去一样健康了。

恢复了精力的杨大佬，开始思考起了一个小问题。为了能正确的描述这个问题，让我们先回顾一些背景知识。

每种丹药都有一个非负整数能力值，杨大佬在炼丹时，每次可以花费一些仙气对某个丹药的能力值进行修改。具体来说，如果有个丹药的能力值是  $x$ ，而杨大佬想将这个丹药的能力值变成  $x \oplus 2^i$ ，则他需要花费的仙气值  $fairy(x, i)$  为：

$$fairy(x, i) = (\max\{x, x \oplus 2^i\})^{2^i} \bmod 19260817 + 1$$

其中  $\oplus$  指按位异或。

现在考虑两个能力值不同的丹药，设他们的能力值分别为  $u$  和  $v$ 。根据上一段关于  $fairy$  函数的描述不难知道，如果杨大佬想将能力值为  $u$  的丹药修改成能力值为  $v$  的丹药，那么他必然是按照一定的顺序轮流修改非负整数  $u$  的每一个二进制位，直到将其修改成  $v$ ，在这一过程中杨大佬花费的总仙气值就是每次修改单一二进制位所花费的仙气值（ $fairy$  值）之和。显然，在将  $u$  修改成  $v$  的所有操作顺序中，应存在一个顺序使得杨大佬花费的总仙气最少，我们记这个最小仙气花费值为  $angel(u, v)$ 。

有了  $fairy(x, i)$  和  $angel(u, v)$  的定义后，我们终于可以毫无偏差地描述杨大佬的小问题了。杨大佬手上有  $n$  个丹药，其中第  $i$  个丹药的能力值为  $a_i$ 。现在，杨大佬想将某个丹药的能力值变成除它本身外的其他某个丹药的能力值。杨大佬想知道的是，他最少需要花费多少仙气才能达成这一目的，也就是说杨大佬想知道下面这个式子的答案：

$$\min_{1 \leq i, k \leq n, i \neq k} \{angel(a_i, a_k)\}$$

你能帮杨大佬解决这个问题吗？

### Input

第一行是一个整数  $n$  ( $1 \leq n \leq 2 \times 10^5$ )，表示杨大佬拥有的丹药个数。

接下来  $n$  行，每行一个整数  $a_i$  ( $0 \leq a_i \leq 2^{18} - 1$ )，给出了每个丹药的能力值。

### Output

输出一行一个整数  $ans$ ，表示  $\min_{1 \leq i, k \leq n, i \neq k} \{angel(a_i, a_k)\}$  的值。

**Example**

standard input	standard output
5 23 76 51 93 42	2071072

## Problem C. 六学家的困惑

Input file:           standard input

Output file:          standard output

小六喜欢两全其美的事情，今天就正好有一个这样的机会。

小六面前有两根管子，管子里面放满了数字为 1 到 9 的小球。每次取球时，小六会先选择一根管子，再从这根管子的某一侧（左侧或右侧）取出一个球。在满足取球规则的情况下，他可以任意顺序取出所有小球。假如小六依次取出的球的编号为  $a_1, a_2, \dots, a_n$ ，则他最后就得到了一个形如  $\overline{a_1 a_2 \dots a_n}$  样的十进制数。

小六希望他的取球顺序所组成的数是最大的，你可以帮一下他吗？

### Input

第一行输入数据组数  $T$  ( $T \leq 14$ )。

接下来  $T$  行，每行输入两串只含 1 到 9 的数字，表示两根管子里的小球的数字。每根管子内至少含 1 个小球，且数量不超过 40 个。

### Output

对每一组数据，输出一行 **Case #x: A**，其中  $x$  是数据组数编号（从 1 开始）， $A$  是小六能组成的最大的数。

### Example

standard input	standard output
2	Case #1: 665544332211
123456 123456	Case #2: 99998888776655498443
9876346789 9854894589	

## Problem D. 神经网络

Input file:        standard input  
Output file:       standard output

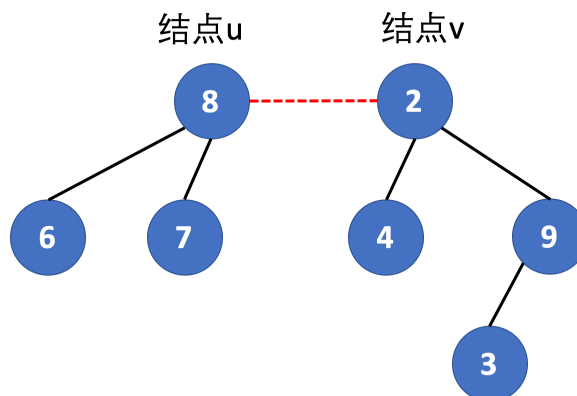
Nowing · 钟喜欢对称的东西，但可惜世界上绝大部分事物都是不对称的，因此 Nowing 精神崩溃了。众所周知，Sega · 黄精通神经网络，因此他打算帮助 Nowing 修复他的神经。

当 Nowing · 钟精神正常时，他的神经可以看作是一棵由  $n$  个神经元组成的树。这  $n$  个神经元编号为  $1 \sim n$ ，且恰由  $n - 1$  条神经纤维连通，每个神经元都有一个神经权重  $a_i$ 。而当 Nowing 精神崩溃后，这  $n - 1$  条神经纤维全部断裂， $n$  个神经元变得两两互不相通。Sega · 黄的工作就是按某种顺序依次修复这  $n - 1$  条神经纤维。

每当 Sega · 黄修复完一条神经纤维，便会有一些新的神经元对相连通，故会产生一些神经脉冲。假设 Sega 准备修复一条连接  $u, v$  神经元的神经纤维，设在修复前  $u$  所在连通块的结点集合为  $G(u)$ ， $v$  所在连通块的结点集合为  $G(v)$ 。那么在修复这条神经纤维后，产生的神经脉冲值计算公式如下：

$$pulse(< u, v >) = \sum_{a \in G(u)} dist(a, v) + \sum_{b \in G(v)} dist(b, u)$$

其中  $dist(a, b)$  指在原本的神经元树上，在不走重复神经元的情况下，由神经元  $a$  走到神经元  $b$  时经过的所有神经元的神经权重之和。



我们来举个例子。考虑上图，图中每个结点上的数值就是它本身的神经权重，其中结点  $u$  的神经权重为 8，结点  $v$  的神经权重为 2。现在要修复图中的虚线边，则修复该边所产生的神经脉冲按如下过程计算：

$$\sum_{a \in G(u)} dist(a, v) = (6 + 8 + 2) + (7 + 8 + 2) + (8 + 2) = 43$$

$$\sum_{b \in G(v)} dist(b, u) = (3 + 9 + 2 + 8) + (9 + 2 + 8) + (4 + 2 + 8) + (2 + 8) = 65$$

$$pulse(< u, v >) = 43 + 65 = 108$$

修复过程中产生的**总神经脉冲量**，是指在修复这  $n - 1$  条神经纤维时，所产生的神经脉冲数值之和。对于 Sega · 黄来说，他每次会从还没有被修复的神经纤维中**随机等概率地**选择一条进行修复，因此最终产生的总神经脉冲量往往是难以预测的。不过 Sega 并不在意这些，他在意的是，最终产生的总神经脉冲量的期望值是多少。

不难证明这个期望值一定是有理数，所以 Sega 只想知道这个期望值在模 998244353 意义下的结果，但他觉得这个问题太简单了，于是就扔给你来做。你能解决这个问题吗？

## Input

第一行一个整数  $n$  ( $1 \leq n \leq 10^5$ ), 表示 Nowing · 钟的神经结点个数。

第二行有  $n$  个整数, 其中第  $i$  个整数  $a_i$  ( $1 \leq a_i \leq 10^5$ ) 表示第  $i$  个神经结点的神经权重。

接下来是  $n - 1$  行, 每行两个整数  $u, v$  ( $1 \leq u, v \leq n$ ), 表示有一条神经纤维连接编号为  $u, v$  的两个神经结点。数据保证给出的神经结点以及神经纤维构成一棵树。

## Output

输出一行一个整数  $ans$  ( $0 \leq ans < 998244353$ ), 表示产生的总神经脉冲量期望值在模 998244353 意义下的结果。也就是说, 如果实际期望值为  $\frac{p}{q}$ , 那么  $ans$  应满足  $ans \times q \equiv p \pmod{998244353}$ 。

## Example

standard input	standard output
4 16 13 8 9 1 2 3 1 3 4	665496476

## Notes

样例的中实际的总神经脉冲量期望值为  $\frac{722}{3}$ , 但因为  $665496476 \times 3 \equiv 722 \pmod{998244353}$ , 故应输出 665496476。



## Problem E. 数独挑战

Input file: standard input

Output file: standard output

数独是一种填数字游戏，英文名叫 Sudoku，起源于瑞士，上世纪 70 年代由美国一家数学逻辑游戏杂志首先发表，名为 Number Place，后在日本流行，1984 年将 Sudoku 命名为数独，即“独立的数字”的缩写，意思是“在每一格只有一个数字”。2004 年，曾任中国香港高等法院法官的高乐德 (Wayne Gould) 把这款游戏带到英国，成为英国流行的数学智力拼图游戏。

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

5	3	4	6	7	8	9	1	2
6	7	2	1	9	5	3	4	8
1	9	8	3	4	2	5	6	7
8	5	9	7	6	1	4	2	3
4	2	6	8	5	3	7	9	1
7	1	3	9	2	4	8	5	6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
3	4	5	2	8	6	1	7	9

一组数独的解

玩家需要根据  $9 \times 9$  盘面上的已知数字，推理出所有剩余位置的数字，并满足每一行、每一列、每一个粗线九宫格内的数字包含有 1-9 的数字，且不重复。

现在给你一个数独，请你解答出来。每个数独保证有且只有一个解。

### Input

输入仅一组数据，共 9 行 9 列，表示初始数独（其中 0 表示数独中的空位）。

### Output

输出共 9 行 9 列，表示数独的解。

注意行末没有空格。

### Example

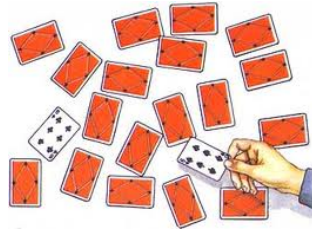
standard input	standard output
5 3 0 0 7 0 0 0 0	5 3 4 6 7 8 9 1 2
6 0 0 1 9 5 0 0 0	6 7 2 1 9 5 3 4 8
0 9 8 0 0 0 0 6 0	1 9 8 3 4 2 5 6 7
8 0 0 0 6 0 0 0 3	8 5 9 7 6 1 4 2 3
4 0 0 8 0 3 0 0 1	4 2 6 8 5 3 7 9 1
7 0 0 0 2 0 0 0 6	7 1 3 9 2 4 8 5 6
0 6 0 0 0 0 2 8 0	9 6 1 5 3 7 2 8 4
0 0 0 4 1 9 0 0 5	2 8 7 4 1 9 6 3 5
0 0 0 0 8 0 0 7 9	3 4 5 2 8 6 1 7 9

## Problem F. 翻牌游戏

Input file: standard input

Output file: standard output

胡老师最近迷上了一款经典小游戏——翻牌。



桌面上有  $n$  个互不相同的对子组成的  $2n$  张扑克牌，背面朝上放在桌面上。

每一轮，胡老师会选择两张不同的牌翻开，如果这两张牌相同，则将它们消除；如果不同，则将它们盖回去。需要注意的是：胡老师总是先选择两张牌，再同时将它们翻开。

胡老师有个外号叫人形 AI，他拥有超强的记忆力，所以 he 可以记住任何他曾经翻开过的牌的位置。他现在想知道，在这种情况下，消除所有牌需要的最少期望轮数是多少。

### Input

第一行是一个整数  $T$  ( $1 \leq T \leq 20000$ )，表示数据组数。

接下来  $T$  行，每行仅包含一个正整数  $n$  ( $1 \leq n \leq 5000$ )，表示一共有  $n$  个对子（即  $2n$  张扑克牌）。

### Output

输出共  $T$  行。

对每组数据，输出一行一个恰有两位小数的浮点数，表示消除所有牌所需最少期望轮数精确到小数点后两位的结果。

### Example

standard input	standard output
1	3.00
2	

### Notes

以下叙述  $n = 2$  时的最优策略。

- (1) 随机选择两张扑克牌。有  $\frac{1}{3}$  概率两张牌相同，直接消除，跳至 (4)；有  $\frac{2}{3}$  概率不同，跳至 (2)。
- (2) 选择一张已经选过的扑克牌，和一张没有选过的扑克牌。有  $\frac{1}{2}$  的概率相同，直接消除，跳至 (4)；有  $\frac{1}{2}$  的概率不同，跳至 (3)。
- (3) 这时已经知道所有扑克牌的图案，选择一个对子消除，跳至 (4)。
- (4) 消除最后两张扑克牌。

数学期望为： $E = \frac{1}{3} \times 2 + \frac{1}{3} \times 3 + \frac{1}{3} \times 4 = 3.00$ 。

## Problem G. 铁树开花

Input file:        standard input  
Output file:       standard output

东东是一名替身使者，但他那个名叫“铁树开花”的替身能力有点无聊——简单来说，“铁树开花”可以使一棵花的种子立刻发芽长大并开花。

现在有一个  $n$  个点  $C$  条边的无向图，图中结点编号为  $1 \sim n$ ，且每条边边权均为 1。在这个图上分布有  $m$  颗花的种子，编号为  $1 \sim m$ ，每颗种子可以被两个参数  $x_i, d_i$ （其中  $d_i \in \{0, 1, 2\}$ ）描述，表示编号为  $i$  的种子埋在了编号为  $x_i$  的结点内，且当其开花时，所有到结点  $x_i$  的最短路长度不超过  $d_i$  的结点都会被这朵花覆盖。

东东可以任意指定一个  $m$  的排列  $p_1, p_2, \dots, p_m$ ，然后依次对编号为  $p_1, p_2, \dots, p_m$  的种子使用“铁树开花”能力，于是无向图中的每一个结点最终都可能被第  $1 \sim m$  中的某一朵花覆盖。需要注意的是，某个结点最终被哪朵花覆盖，取决于最后覆盖它的是哪朵花。比如在执行整个开花顺序的过程中，结点  $u$  先后被编号为 4, 8, 2 的花覆盖，那么我们称结点  $u$  最终是被编号为 2 的花覆盖的。我们记最终被编号为  $i$  的花覆盖的结点的个数为  $cnt_i$ 。

东东很无聊，他想找到这样的一个排列，使得按照这个排列使用“铁树开花”能力后，能达成下面这个目标：

$$\max\{\min_{1 \leq i \leq m}\{cnt_i\}\}$$

也就是说，要找到一个排列  $P$ ，使得在考虑每一个可能的排列下最小的那个  $cnt_i$  时，排列  $P$  对应的那个  $\min_{1 \leq i \leq m}(cnt_i)$  是最大的。显然这样的排列  $P$  可能有很多个，所以东东想要找出在所有满足上述条件的排列  $P$  中，字典序最小的那个。

东东自然是一眼看穿了答案，于是把这个问题扔给你来做。他为了不让你难过，保证了给出的无向图一定是一棵树。他希望你求出  $\max\{\min_{1 \leq i \leq m}\{cnt_i\}\}$ ，以及在能达成这一目标且字典序最小的那个排列下，每朵花的  $cnt_i$  值。

### Input

第一行两个整数  $n, m$  ( $1 \leq n \leq 10^5, m \leq 2 \times 10^4$ )，分别表示树的结点数和种子个数。

接下来  $n - 1$  行给出了树的结构，每行两个整数  $u, v$  ( $1 \leq u, v \leq n, u \neq v$ )，表示树中有一条连接结点  $u, v$  的边。

最后是  $m$  行，其中第  $i$  行是两个整数  $x_i, d_i$  ( $1 \leq x_i \leq n, d_i \in \{0, 1, 2\}$ )，表示第  $i$  颗种子（第  $i$  朵花）的参数，意义如题目描述中所述。

### Output

输出共  $m + 1$  行。

第一行输出  $\max\{\min_{1 \leq i \leq m}\{cnt_i\}\}$  的值。

接下来  $m$  行，每行输出一个整数，其中第  $i$  行为在满足条件且字典序最小的开花顺序下的  $cnt_i$  值。

**Example**

standard input	standard output
10 3	1
3 1	3
5 2	1
2 4	4
1 2	
6 3	
3 7	
4 8	
10 7	
9 5	
5 2	
2 0	
3 1	

## Problem H. Parco\_Love\_GCD

Input file:        **standard input**

Output file:      **standard output**

众所周知，在算法竞赛中，出题人对他出的题的难度往往存在错误的估计。比如出题人本想出个中等题，没想到却出成了简单题；本想出个自闭题，结果数据太水变成了签到题。因此，为了让一场比赛能有良好的体验，有一个靠谱的验题人是非常重要的。

CC 出好题目后，便拿给小马哥看。不出所料，这些题目小马哥全都是看一眼就会做了。而且，小马哥觉得这些题对于参赛选手来说也太水了（5 个签到题哦～）。为了避免发生全场十几个队 AK 这种紧急事态，小马哥决定把一道签到题改成简单题，于是便有了现在这个题目。

小马哥非常喜欢数论，尤其钟爱“最小公倍数”（Greatest Common Divisor，简称 GCD）这一概念，因此他打算出一道和 GCD 有关的题目。小马哥首先给出了含  $n$  个正整数的序列  $a_1, a_2, \dots, a_n$ ，然后让你考虑该序列的所有子区间的数对应的 GCD 值，也就是说考虑所有  $\gcd(a_l, \dots, a_r)$  的值。显然，这样的值一共有  $\frac{n(n+1)}{2}$  个。一个中二出题人可能会让你求这  $\frac{n(n+1)}{2}$  数中第  $k$  大的数，但幸运的是，小马哥是个正经出题人，因此它只让你求这  $\frac{n(n+1)}{2}$  个数之和模  $10^9 + 7$  的结果。

也就是要求下面这个式子：

$$ans = \left( \sum_{l=1}^n \sum_{r=l}^n \gcd(a_l, \dots, a_r) \right) \bmod (10^9 + 7)$$

小马哥在此预祝大家 AK ～

### Input

第一行是一个整数  $n$  ( $1 \leq n \leq 5 \times 10^5$ )，表示数的个数。接下来一行给出  $n$  个整数，其中第  $i$  个整数  $a_i$  满足  $1 \leq a_i \leq 10^9$ 。

### Output

输出一行一个整数，表示  $ans$  的值。

### Example

standard input	standard output
5 16 4 7 21 3	72

## Problem I. 炒股

Input file:           standard input

Output file:          standard output

攒机一时爽，一直攒机一直爽。

沉迷攒机的胡老师很快就发现，他每天只能靠吃泡面过活了。为了改善伙食，同时继续攒机大业，胡老师决定下海炒股。

胡老师有特别的炒股技巧。首先他会选定一支他看好的股票，然后永远只买这一支股票。此外，胡老师每天要么只买入股票要么只卖出股票，且出于某种不为人知的原因，胡老师手上最多只能持有 1 股的股票。胡老师每天会根据当天的股价及手上的持股数决定是买入还是卖出股票，需要注意的是，只要胡老师选择了买入或卖出，那么一定可以按当天的价格买入或卖出股票。

炒股需要本金，但胡老师的钱都拿去攒机了，于是他去找 CC 借了一大笔钱（这笔钱可以视为无穷多），并约定  $n$  天之后归还。另一方面，为最大化  $n$  天内的收益，胡老师通过不为人知的 py 交易获取了接下来  $n$  天每天的股票价格。

在一开始，胡老师手上没有持有股票。胡老师想知道，在  $n$  天之后他最多能靠炒股赚多少钱。

### Input

第一行是一个整数  $n$  ( $1 \leq n \leq 5 \times 10^5$ )，表示天数。

接下来  $n$  行给出了接下来  $n$  天内胡老师看好的股票每天的价格，其中第  $i$  行是一个整数  $a_i$  ( $1 \leq a_i \leq 10^6$ )，表示该股票在第  $i$  天的价格。

### Output

输出一行一个整数  $ans$ ，表示  $n$  天之后胡老师最多能赚的钱数。

### Example

standard input	standard output
5 1 3 2 6 4	6

## Problem J. 单身狗救星

Input file: standard input

Output file: standard output

为了提高华南理工大学的脱单率，沈老师办了个相亲交友活动（具体形式可以参考非诚勿扰），面向广大华工学子。

参加活动的每一位同学都有一个帅气/美丽程度  $B$  以及一个聪明程度  $I$ 。或许你会觉得，找对象自然是应该找越好看且越聪明的，但对万年单身的华工学子来说，他们只想找最适合自己的。考虑两位同学 Alice 和 Bob。Alice 的帅气/美丽程度是  $B_1$ ，聪明程度是  $I_1$ ；Bob 的帅气/美丽程度是  $B_2$ ，聪明程度是  $I_2$ 。则 Alice 和 Bob 作为伴侣来说的适合度按如下公式计算：

$$\frac{\begin{vmatrix} B_1 & I_2 \\ B_2 & I_1 \end{vmatrix} + \begin{vmatrix} B_2 & I_2 \\ B_1 & I_1 \end{vmatrix}}{\begin{vmatrix} B_1 & B_2 \\ B_2 & B_1 \end{vmatrix}}$$

现在，台上有  $n$  位男同学（类似非诚勿扰的规则），编号为  $1 \sim n$ ，他们的帅气程度与聪明程度都是已知的。台下有  $m$  位女同学，每位女同学也有自己的美丽度与聪明度，而她们都想从台上的  $n$  位男同学中选择一个最适合自己的对象，也就是说想找一个对自身来说适合度最大的对象。如果有多位男同学的适合度都是最大的，则选择编号较小的做对象。

沈老师想知道，台下的每位女同学，究竟打算选谁做为自己的对象。需要注意的是，由于现在还只是思考阶段，没有正式开始选对象过程，所以有多位女同学选择同一位男同学是可能的。

### Input

第一行是两个整数  $n, m$  ( $1 \leq n, m \leq 10^5$ )，分别表示男同学人数及女同学人数。

接下来  $n$  行每行两个整数，其中第  $i$  行是  $B_i, I_i$  ( $-10^9 \leq B_i, I_i \leq 10^9$ )，表示表示第  $i$  位男同学的帅气程度及聪明程度。数据保证不存在两位男同学，他们的帅气及美丽程度完全相同。此外，也不存在这样的三位男同学  $a, b, c$ ，使得  $a$  和  $b$  的适合度与  $b$  和  $c$  的适合度相同。

接下来  $m$  行每行两个整数，其中第  $i$  行是  $B'_i, I'_i$  ( $-10^9 \leq B'_i, I'_i \leq 10^9$ )，表示表示第  $i$  位女同学的美丽程度及聪明程度。数据保证不存在一名女同学与一名男同学，他们的聪明程度的绝对值相同。

### Output

输出共  $m$  行，每行一个整数，其中第  $i$  行表示第  $i$  位女同学打算选择的男同学的编号。

### Example

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

## Problem K. Parco\_Love\_String

Input file: standard input

Output file: standard output

众所周知，在算法竞赛中，出题人对他出的题的难度往往存在错误的估计。比如出题人本想出个简单题，没想到却出成了重坑细节题；本想出个中等难度题，结果却变成了防 AK 题。因此，为了让一场比赛能有良好的体验，有一个靠谱的验题人是非常重要的。

CC 出好题目后，便拿给小马哥看。不出所料，这些题目小马哥全都是看一眼就会做，但他觉得这对于参赛选手来说还是有点难。为了避免 CC 被喷成毒瘤出题人，小马哥准备加一道签到题。

小马哥有一个只由小写字母组成的字符串  $S$ ，他会问你一些关于这个字符串的问题。小马哥每次提问时会给你一个整数  $i$ ，意思是要把字符串  $S$  在第  $i$  和第  $i+1$  个字符之间切断，这样便得到两个新串  $S[1, i]$  及  $S[i+1, |S|]$ 。记  $A = S[1, i]$ ， $B = S[i+1, |S|]$ 。现在，先考虑  $A$  的所有连续子串，再考虑  $B$  的所有连续子串，小马哥问你  $A, B$  之间相同子串对有多少个。

也就是说，小马哥想让你计算下面这个式子：

$$ans = \sum_{1 \leq l_1 \leq r_1 \leq |A|} \sum_{1 \leq l_2 \leq r_2 \leq |B|} [A[l_1, \dots, r_1] == B[l_2, \dots, r_2]]$$

其中，求和式右边的式子是这样一个函数：

$$[x] = \begin{cases} 1 & (x = True) \\ 0 & (x = False) \end{cases}$$

你能解决小马哥的问题，并签到成功吗？

### Input

第一行是一个字符串  $S$ ，保证只由小写字母组成，且  $1 \leq |S| \leq 1000$ 。

第二行是一个正整数  $T$  ( $1 \leq T \leq 10^5$ )，表示询问次数。

接下来  $T$  行，每行一个整数  $i$  ( $1 \leq i \leq n-1$ )，表示一个询问。

### Output

对于每个询问，输出一行一个整数表示对应询问的  $ans$ ，意义如题目描述中所述。

### Example

standard input	standard output
ababa	2
4	4
1	4
2	2
3	
4	

### Notes

对于  $i = 3$  这个询问，原串被拆分成  $A = aba$ ， $B = ba$ 。

此时  $A$  的所有子串为： $a$ ， $a$ ， $b$ ， $ab$ ， $ba$ ， $aba$ ； $B$  的所有子串为： $a$ ， $b$ ， $ba$ 。

因此  $A, B$  之间相同子串对为： $(a, a)$ ， $(a, a)$ ， $(b, b)$ ， $(ba, ba)$ ，共计 4 个。

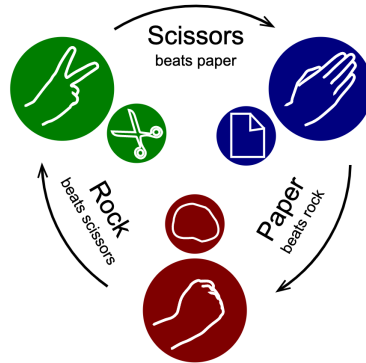


## Problem L. 石头剪刀布

Input file:        standard input

Output file:      standard output

石头剪刀布是一个人人都喜欢玩的休闲小游戏，其制胜规则如下图。



图片来源：Wikipedia

有研究表明，在玩石头剪刀布时，人们大多倾向于“赢家保持现状输家做出改变的策略”。这意味着，如果你是输家，下一轮应该用能打败对手的出手；如果你是赢家，下一轮就不要再使用原来的出手。

对于 CC 来说，这些心里分析都是无所谓的，因为他是灵能力者，可以预知对手想要出石头（Rock），剪刀（Scissors），还是布（Paper）。CC 想要每轮比赛都赢，请你告诉 CC 他每轮应该出什么才能赢。

### Input

第一行是一个整数  $n$  ( $1 \leq n \leq 100$ )，表示比赛轮数。

接下来  $n$  行，每行一个字符串。这个字符串只能是 Rock，Scissors，Paper 三者之一，表示本轮对手的出招是石头还是剪刀还是布。

### Output

输出共  $n$  行。

对对手的每个出招，输出一行一个字符串。这个字符串只能是 Rock，Scissors，Paper 三者之一，表示本轮 CC 需要出石头还是剪刀还是布才能赢对手。

### Example

standard input	standard output
3	Rock
Scissors	Paper
Rock	Scissors
Paper	