

杭州师范大学第二届程序设计新生赛

计算机科技协会

December, 22, 2019

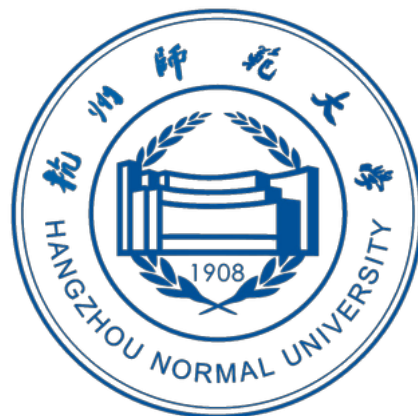


杭州师范大学
HANGZHOU NORMAL UNIVERSITY

Problems

A	KK 画猪
B	KK 学几何
C	KK 算日期
D	KK 与卡牌
E	KK 与答辩
F	KK 与刷题
G	KK 看跳舞
H	KK 与十佳
I	KK 买股票
J	KK 与英语
K	KK 与线代
L	KK 学五子棋
M	KK 与二叉树

Do not open before the contest has started.



Problem A. KK 画猪

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

猪年临近尾声，kk 想画一只猪纪念一下过去的一年，但是他的美术水平太差了，请您帮他画一只猪吧！

Input

“pig”.

Output

一只猪。

Example

standard input	standard output
pig	(____/) / @__@ \ ((oo)) '-.~~.-' / \ @/ _ (/ / \ \) WW'-----'WW

Problem B. KK 学几何

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

kk 最近迷上了几何学，对计算各种图形的面积很有兴趣，现在他想来考考你。

已知 kk 会考你三种图形，分别是圆，三角形和矩形，分别用 ‘1’，‘2’，‘3’ 来表示三种图形：

- 如果是 1，则会告诉你圆的半径 r 。
- 如果是 2，则会告诉你三角形的底 L 和高 h 。
- 如果是 3，则会告诉你矩形的长 L 和宽 W 。

现在希望你把 kk 给你的所有图形的面积累加起来，并保留 1 位小数。

并且我们约定 $\pi = 3$ 。

Input

第一行包含一个整数 $n(1 \leq n \leq 10^3)$ ，代表有 n 个几何图形需要你计算。

接下来有 n 行，每行开头都是一个数字 $t(1 \leq t \leq 3)$ ，表示图形的类型，1 代表圆，2 代表三角形，3 代表矩形。

- 如果是 1，则代表圆，后面跟着一个整数，表示半径 r 。
- 如果是 2，则代表三角形，后面跟着两个整数，表示 L 和 h 。
- 如果是 3，则代表矩形，后面跟着两个整数，表示 L 和 W 。

注意，和图形有关的所有尺寸都在 100 以内。

Output

输出只有一行，代表所有图形面积总和，注意保留一位小数。

Example

standard input	standard output
3 2 6 8 1 8 3 6 2	228.0

Note

- 圆的面积 $= \pi \cdot r^2$
- 三角形的面积 $= \frac{L \cdot h}{2}$
- 矩形的面积 $= L \cdot w$

Problem C. KK 算日期

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

2019年的二月只有 28 天，2020 年的二月有 29 天啊!

kk

大学霸 kk 一秒算出了明年是个闰年（闰年的定义在 Note 中有解释），二月有 29 天。

那若干年后的二月会有几天呢？

大学霸 kk 的脑细胞可不能浪费在算日子上，这种粗活就交给你叭！

已知今年为公元 n 年，kk 想知道 δ 年后的二月有多少天，请你帮他算一算叭！

(为了简化，我们假设公元前的日期直接取绝对值运算，例如公元前 123 年，当作公元 123 年来计算)

Input

第一行一个整数 $T(1 \leq T \leq 10^3)$ 表示有 T 组输入。

每组输入两个整数 $n, \delta(0 < n \leq 10^4, -10^4 \leq \delta \leq 10^4)$ ，表示当前的年份和 kk 想知道 δ 年后的二月有几天（显然当 $\delta < 0$ 时表示 δ 年前）。

Output

对于每组输入，输出一个整数 x ，表示那一年二月有 x 天。

Example

standard input	standard output
3	29
2019 1	28
2019 -1	29
2019 -3	

Note

2019 往后一年是 2020 年，是闰年，29 天

2019 往前一年是 2018 年，是平年，28 天

2019 年前三年是 2016 年，是闰年，29 天

公元 0 年不存在！不存在！不存在！

闰年是公历中的名词，分为普通闰年和世纪闰年。

普通闰年：公历年份是 4 的倍数的，且不是 100 的倍数，为闰年。（如 2004 年就是闰年）。

世纪闰年：公历年份是整百数的，必须是 400 的倍数才是闰年（如 1900 年不是世纪闰年，2000 年是世纪闰年）。

——源自百度百科

Problem D. KK 与卡牌

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

在全新的卡牌世界里，每一张卡牌都代表了一个英雄。

英雄有自己的名字 s 和武力值 w ，英雄的名字是一个字符串，武力值是一个浮点数。

- 武力值越高，则代表英雄的武力越强。
- 在武力相同的情况下，名字的字典序越靠前，则能力越强。

kk 拥有 n 张卡牌，而 bm 有 q 张卡牌，kk 想知道对于 bm 的每一张卡牌，自己的 n 张卡牌中总共有几张卡牌能够战胜他。

战胜的条件是：

- 能力值比他高。
- 或者在能力值相同的情况下英雄名字的字典序比他靠前，如果能力值和名字完全一样，则平手。

字典序的定义在下方的 Note 中。

Input

第一行是一个整数 $n(1 \leq n \leq 10^5)$ ，代表 kk 拥有 n 张卡牌。

接下来有 n 行，每行有一个字符串 $s(1 \leq |s| \leq 5)$ ，和一个浮点数 $w(0 \leq w \leq 10^2)$ 。

第 $n+2$ 行是一个 $q(1 \leq q \leq 10^5)$ ，代表 bm 拥有 q 张卡牌。

接下来有 q 行，每行有一个字符串 $s(1 \leq |s| \leq 5)$ ，和一个浮点数 $w(0 \leq w \leq 10^2)$ 。

s 和 w 的含义如题目中描述，名字均为小写字母， w 均为1位小数。

名字可能会有重复。

Output

对于 bm 的每张卡牌，都输出一个整数，代表 kk 的卡牌中，能打败 bm 的卡牌的数量，若没有能打败 bm 的卡牌，则输出 0。

Example

standard input	standard output
5	3
tom 3.5	2
joe 1.0	5
jark 2.1	1
emily 1.5	
kk 1.2	
4	
kk 1.2	
kk 1.6	
aa 0.9	
aa 2.2	

Note

不建议在不关同步的情况下，用 cpp 的 `cin` 和 `cout` 做读入和输出，可能会导致超时。

字典序定义：

按照第一个字母、以 a、b、c...z 的顺序排列；如果第一个字母一样，那么比较第二个、第三个乃至后面的字母。如果比到最后两个单词不一样长（比如，`sigh` 和 `sight`），那么把短者排在前。（来自百度百科）

Problem E. KK 与答辩

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

kk 经常参加答辩，最近，他一共参加了 n 场答辩。

每场答辩有 a_i 个人，每个人的总分有两个组成部分，一个是评委打分，一个是观众打分，两者相加就是这个人在这场答辩的总分。

现在，如果有这样一个同学（一个同学可以参加多场答辩，名字相同视为同一人）：在他参加的所有答辩里，总分都低于 kk，我们称他被 kk 碾压了。

请聪明的你帮忙算一下有几位可怜的同学被 kk 碾压了。

Input

第一行为 $T(1 \leq T \leq 10)$ ，表示有 T 组输入。

每组输入第一行是一个整数 $n(1 \leq n \leq 10)$ 。

接下来分为 n 个部分，每个部分会有一个整数 $a_i(1 \leq a_i \leq 10)$ ，表示第 i 场答辩有 a_i 个人。

接下来的 a_i 行：

- 每行有一个字符串 $s(1 \leq |s| \leq 10)$ ，都是小写字母，不含空格)，表示答辩的同学的名字。
- 两个整数 $b_i, c_i(1 \leq b_i, c_i \leq 50)$ ，表示评委打分和观众打分。
- 我们约定第一行的数据表示 kk 的名字和他的分数。

Output

输出一个整数，表示被 kk 碾压的同学数量。

Example

standard input	standard output
1 1 2 kk 50 50 bm 1 1	1

Problem F. KK 与刷题

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

传言道：失败是成功之母。

想必大家在做题目的时候经常会出现 WA, TLE, MLE 等等问题，这些问题在 kk 身上依旧会发生。

这些年的竞赛时间中，kk 也是 WA 了无数次的男人。

但 kk 有一个习惯，每天都会记录下来他 WA 的次数，WA 的越多，他就认为他做的题目越来越难，自己的能力值也越来越高。

相反，如果他做题一路顺畅，那么他认为他的能力并没有被锻炼，自己的能力值反而会减少，具体计算能力值的方法如下：

- 之前每有一天的 WA 的次数比今天多，都会使他的能力值 -1 。
- 之前每有一天的 WA 的次数比今天少，都会使他的能力值 $+1$ 。
- 之前每有一天的 WA 的次数与今天相同，不会影响他的能力值。

每天的能力值将继承至下一天。初始能力为 0。

因为第一天没有 WA 的次数可以比较，所以第一天结束后，能力值不会改变。

求 kk 能力值最高时为多少，以及当前 kk 的能力值的大小

Input

第一行输入一个数 $n(1 \leq n \leq 10^5)$ ，代表 kk 已经训练的天数。

第二行输入 n 个数字 $a_i(1 \leq a_i \leq 10^5)$ ，表示 kk 每天的 WA 的次数。

Output

输出两个数，第一个数字代表 kk 能力值最高为多少，以及当前 kk 的最终能力值。

Examples

standard input	standard output
3 1 3 2	1 1
3 2 1 3	1 1
4 1 5 4 1	1 -1
10 4 3 1 5 6 7 8 9 10 2	30 23

Note

样例1:

- 第一天 kk 能力值为 0。
- 第二天 kk WA 了 3 次，比第一天多，那么他的能力值加一，为 1。
- 第三天 kk WA 了 2 次，比第一天多，那么他的能力值加一，比第二天少，那么他的能力值减一，最后还是为 1。
- 三天中能力值最大的时候为 1。

样例2:

- 第一天 kk 能力值为 0。
- 第二天 kk WA 了 1 次，比第一天少，那么他的能力值减一，为 -1。
- 第三天 kk WA 了 3 次，比第一天多，能力值加一，比第二天多，能力值有加一，那么最后他的能力值为 1。
- 三天中能力值最大的时候为 1。

样例3:

- 第一天 kk 能力值为 0。
- 第二天 kk WA 了 5 次，比第一天多，那么他的能力值加一，为 1。
- 第三天 kk WA 了 4 次，比第一天多，能力值加一，比第二天少，能力值减一，因此能力值仍为 1。
- 第四天 kk WA 了 1 次，和第一天一样，能力值不变，比第二，三天都少，能力值减二，因此能力值为 -1。
- 四天中能力值最大的时候为 1。

样例四:

- 第一天 kk 能力值为 0。
- 第二天 kk WA 了 3 次，比第一天少，那么他的能力值减 1，为 -1。
- 第三天 kk WA 了 1 次，比第一，二天少，那么他的能力值减 2，为 -3。
- 第四天 kk WA 了 5 次，比第一二三天多，那么他的能力值加 3，为 0。
- 第五天 kk WA 了 6 次，比第一二三四天多，那么他的能力值加 4，为 4。
- 第六天 kk WA 了 7 次，比第一二三四五天多，那么他的能力值加 5，为 9。
- 第七天 kk WA 了 8 次，比第一二三四五六天多，那么他的能力值加 6，为 15。
- 第八天 kk WA 了 9 次，比第一二三四五六七天多，那么他的能力值加 7，为 22。
- 第九天 kk WA 了 10 次，比第一二三四五六七八天多，那么他的能力值加 8，为 30。
- 第十天 kk WA 了 2 次，比第三天多，比第一二四五六七八九天少，那么他的能力值减 7，为 23。
- 十天中能力最大为 30。

Problem G. KK 看跳舞

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

一天 kk 发现有 n 个人正围着圈跳舞，并且他们每个人身上都贴着一个范围在 $[1, n]$ 且不重复的数字。

kk 觉得他们在跳「名字没有什么意义但是特别长」舞，这种舞有一种显著的特征就是每个人都穿一件带有号码的衣服，并且按逆时针或者顺时针的顺序排开，绕着人群中心旋转。

kk 看到他们的时候他们已经跳了一会儿了，kk 细心地记下了从他面前跳过的人的编号。

他想知道他们跳的到底是不是「名字没有什么意义但是特别长」舞，现在 kk 把从眼前跳过的每个人的序号告诉你，请你告诉他他们跳的是不是 kk 心中的舞步。

Input

第一行一个整数 $T(1 \leq T \leq 10)$ 代表有 T 组输入。

每组输入包含两行。

- 第一个行一个数字 $n(1 \leq n \leq 10^4)$ ，代表有 n 个人从 kk 眼前跳过。
- 第二行 n 个数字表示从 kk 眼前跳过的每个人的编号，注意这 n 个数字为 $1 \sim n$ 的排列，即 $1 \sim n$ 每个数字出现且只出现一次。

Output

- 如果他们跳的是「名字没有什么意义但是特别长」舞，请输出 ‘YES’(没有引号)。
- 否则输出 ‘NO’(没有引号)。

Example

standard input	standard output
5	YES
3	YES
1 2 3	NO
3	YES
1 3 2	YES
4	
1 3 2 4	
1	
1	
5	
3 2 1 5 4	

Problem H. KK 与十佳

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

kk 在十佳歌手比赛现场唱了一首歌。

底下有 n 名评委，分别给 kk 打出了分数（不包含0，不重复）。

现在 kk 有一个权力，可以把一个评委打的分数去掉，之后其余剩下的所有评委的分数的乘积就是 kk 的分数。

kk 希望自己的分数尽可能高，请问他应该删除哪一个分数呢。

Input

单组输入。

输入一个 $n(2 \leq n \leq 3 \cdot 10^4)$ ，代表评委的个数。

接下来输入 n 个整数 $a_i(a_i \in [-1000000, 0) \cup (0, 1000000])$ ，代表 n 名评委打的分数。

Output

输出一个数，即删除的那个数。

Examples

standard input	standard output
5 1 2 3 4 5	1
3 -6 8 9	-6
5 100 101 102 103 104	100

Note

样例 1：删除了 1 之后，kk 的得分为 $2 \cdot 3 \cdot 4 \cdot 5 = 120$ 。

样例 2：删除 -6 之后，kk 的得分为 $8 \cdot 9 = 72$ 。

样例 3：删除 100 之后，kk 的得分为 $101 \cdot 102 \cdot 103 \cdot 104$ 。

Problem I. KK 买股票

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

kk 在股票模拟软件里看到了一只名为「蛇蛇股份」的股票，他觉得这只股票很有潜力。

kk 有一个能力，他可以看到接下来 n 天里「蛇蛇股份」的价格，因此他也受到了限制，在这 n 天他最多只能买入和卖出一次股票（他卖股票的前提条件是持有股票），请聪明的你帮 kk 计算一下他在这 n 天最多可赚多少钱

kk 初始的时候拥有无限多的钱。

Input

第一行是一个整数 $n(2 \leq n \leq 10^5)$ 。

第二行有 n 个整数,第 i 个整数 $p_i(1 \leq p_i \leq 10^3)$ 表示第 i 天一手蛇蛇股份的价格。

Output

输出 kk 最多可以赚的钱(如果没有赚到钱的方案，即不管怎么样都会亏钱，就输出 0)。

Examples

standard input	standard output
6 10 1 7 3 8 6	7
3 8 3 2	0

Note

样例 1 中，在第二天买入一手，第五天卖出一手，赚得 $8 - 1 = 7$ 元。

样例 2 中，kk 不管在哪天买入，再卖都赚不到钱，输出 0。

Problem J. KK与英语

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

作为大学霸的 kk 轻而易举的解决了前一个问题，还想出了三种解法，并直呼“Too Simple!!!”突然 qq 的窗口抖动了一下，有人发了一个问题过来，大学霸 kk 一看傻了眼，怎么是英语...原来，大学霸 kk 最薄弱的科目就是英语了。

kk 定睛一看：“还好这题目不难，就是时态用错了嘛！”

原来问题目的人把句子的时态弄混了，所有的“was”都被他写成了“is”。

现在他希望 kk 能够帮他纠正这些错误。kk 觉得这种题目太侮辱他的英语水平了，于是他反手把这道题扔给了你。

现在我们将简化问题为如下：只需将一行话中单独的“is”(没有引号)（即“is”前后都有空格）全部改成“was”(没有引号)即可。

Input

一行字符串 $S(1 \leq |S| \leq 10^5)$ 。

Output

输出一行修改后的字符串。（行末无多余空格）

Examples

standard input	standard output
It is my boy friend.	It was my boy friend.
Is it your pencil?	Is it your pencil?
This is an island.	This was an island.
is	is
A is B is C	A was B was C

Note

对于 sample2, 字符串开头为“Is”而不是“is”，所以不替换，直接输出。

对于 sample4, “is”的前后没有空格，不替换，直接输出。

对于 sample5, 一共有两个符合条件的“is”，全部替换。

请注意，输入的句子不一定符合语法，你只需要将所有前后有空格的“is”全部替换为对应的“was”

Problem K. KK 与线代

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

kk 是个乐于助人的学霸，经常有各种各样的人问他各种各样的问题。

众所周知

一天，一个学妹给 kk 发来了这么一道题：

当 x 取 $[l, r]$ 的整数时，求下面行列式的最小值。

$$\begin{vmatrix} 6 & 5 & -1 & -3 \\ 3 & x & 6 & 7 \\ 4 & -5 & 7 & 8 \\ 4 & 3 & 9 & 1 \end{vmatrix}$$

Input

输入为一行，包含两个整数 $l, r, (l \leq r \leq 100000)$ 。

Output

一个整数，表示最小值。

Example

standard input	standard output
1 1	-2470

Note

行列式的值 $D = \sum (-1)^{\tau(k)} a_{1k_1} a_{2k_2} a_{3k_3} \cdots a_{nk_n}$

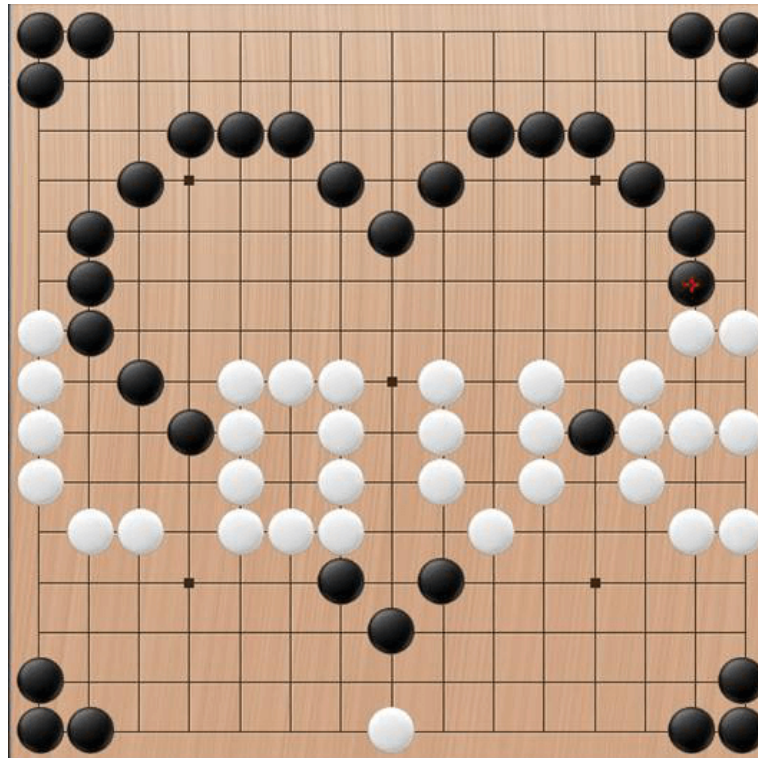
其中序列 k 是将 $1 \sim n$ 排列后组成的序列，共 $n!$ 个。

$\tau(k)$ 表示序列 k 的逆序数。

序列 k 的逆序数表示 k 中 $k_i > k_j$ 且 $i < j$ 的数对个数。

Problem L. KK 学五子棋

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



有一天 kk 一个人在研究残局，现在他得到了一个五子棋的残局，当前轮到黑色方下子，他在思考：

- 他能否在这一步内胜利？如果能，那么他需要把棋子下在哪儿？
- 他若在这一步内不能胜利，那么他把棋子下在哪儿，使得对方不能在下一步获胜？

此处我们约定，五子棋获胜条件为：

当某一方落子后，这一方有大于等于 5 颗相同颜色的棋子在一条线上(可以为横连、纵连、斜连)，那么该方获胜。

Input

多组数据评测。

第一行包含一个正整数 $T(1 \leq T \leq 10^2)$ ，表示有 T 组数据。

对于每一组数据：

第一行包含两个正整数 $n, m(5 \leq n, m \leq 10^3)$ ，表示棋盘的大小。

接下来 n 行，每行 m 个整数 $a_i(0 \leq a_i \leq 2)$ ，表示该位置的棋子状态。

- 0 表示该位置为空
- 1 表示该位置为白棋

- 2 表示该位置为黑棋

数据保证 $\sum n \cdot m \leq 10^6$ 。

数据保证当前局面下白棋和黑棋的数量一样多。

数据保证棋盘上 0 的个数大于等于 2。

数据保证当前局面是非胜利局面，即没有五个白棋连子，也没有五个黑棋连子。

Output

如果 KK 能够在这一步内胜利(输出包含两行):

第一行输出 ‘Win’(没有引号)。

第二行输出包含两个以空格分割的正整数 $x\ y(1 \leq x \leq n, 1 \leq y \leq m)$ ，表示 KK 在这一步要下的棋子的位置。

如果有多个解，那么输出 x 最小的解，如果还有多解，输出 y 最小的解。

如果 KK 不能够在这一步内胜利，但是可以使得对方不能在下一步内胜利(输出包含两行):

第一行输出 ‘Defense’(没有引号)。

第二行输出包含两个以空格分割的正整数 $x\ y(1 \leq x \leq n, 1 \leq y \leq m)$ ，表示 KK 在这一步要下的棋子的位置。

如果有多个解，那么输出 x 最小的解，如果还有多解，输出 y 最小的解。

否则，kk 在这一步不管怎么下，对方都可以在下一步胜利(输出包含一行):

第一行输出 ‘Defeated’(没有引号)。

Example

standard input	standard output
2	Defense
5 6	1 2
0 0 0 0 2 0	Win
0 0 1 0 2 2	3 6
0 0 0 1 0 2	
0 0 0 0 1 0	
0 0 0 0 0 1	
5 10	
1 1 0 2 0 0 0 2 0 0	
1 1 0 0 2 0 2 0 0 0	
1 1 0 0 0 0 0 0 0 0	
1 1 0 0 2 0 2 0 0 0	
0 0 0 2 0 0 0 2 0 0	

Problem M. KK 与二叉树

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

众所周知，先序遍历序列和中序遍历序列可以唯一的确定一棵二叉树，或者后序遍历序列和中序遍历序列也可以唯一的确定一棵二叉树。

不幸的是，先序遍历序列和后序遍历序列却不一定能够唯一的确定一棵二叉树。

更不幸的是，现在 kk 得到了一个先序遍历序列和一个后序遍历序列，它想知道这个先序遍历序列和后序遍历序列能否唯一的确定一棵二叉树。

- 如果能够唯一确定一棵二叉树，那么你需要输出该二叉树的中序遍历序列。
- 如果不能够唯一确定一棵二叉树，那么输出所有可能的二叉树的中序遍历序列结果中字典序最小的一个。

Input

多组数据评测。

第一行一个正整数 $T(1 \leq T \leq 10^5)$ ，表示数据组数。

对于每组数据：

- 第一行一个正整数 $n(1 \leq n \leq 10^5)$ ，表示二叉树的结点个数。
- 第二行 n 个互不相同的正整数 $a_i(1 \leq a_i \leq 10^9)$ ，表示该二叉树的先序遍历序列。
- 第三行 n 个互不相同的正整数 $b_i(1 \leq b_i \leq 10^9)$ ，表示该二叉树的后序遍历序列。

数据保证 $\sum n \leq 10^5$ 。

Output

输出包含 T 组，对于每一组数据：

- 如果给出的先序遍历序列和后序遍历序列能够唯一确定一棵二叉树：
 1. 第一行输出一个 ‘Yes’(没有引号)。
 2. 第二行输出该二叉树的中序遍历序列。
- 否则：
 1. 第一行输出一个 ‘No’(没有引号)。
 2. 第二行输出所有可能的二叉树的中序遍历序列结果中字典序最小的一个。

Example

standard input	standard output
2	Yes
7	2 1 6 4 7 3 5
1 2 3 4 6 7 5	No
2 6 7 4 5 3 1	2 1 3 4
4	
1 2 3 4	
2 4 3 1	

Note

字典序: 如果有两个长度相同(假定长度为 n)的序列 a_i 和 b_i , 如果存在一个 $x \in [1, n]$, 满足:

$$a_1 = b_1, a_2 = b_2, \dots, a_{x-1} = b_{x-1} \quad \text{and} \quad a_x < b_x$$

这两个条件, 那么我们认为序列 a_i 的字典序比序列 b_i 的字典序要小。

二叉树: 二叉树是每个结点最多有两个子树的树结构。

先序遍历序列(preorder traversal sequences): 根据根左右的顺序沿一定路径经过路径上所有的结点得到的遍历序列。

遍历方法: 首先访问根结点然后遍历左子树, 最后遍历右子树。在遍历左、右子树时, 仍然先访问根结点, 然后遍历左子树, 最后遍历右子树, 如果二叉树为空则返回。

中序遍历序列(inorder traversal sequences): 根据左根右的顺序沿一定路径经过路径上所有的结点得到的遍历序列。

遍历方法: 中序遍历首先遍历左子树, 然后访问根结点, 最后遍历右子树。若二叉树为空则结束返回

后序遍历序列(postorder traversal sequences): 根据左右根的顺序沿一定路径经过路径上所有的结点得到的遍历序列。

遍历方法: 后序遍历首先遍历左子树, 然后遍历右子树, 最后访问根结点, 在遍历左、右子树时, 仍然先遍历左子树, 然后遍历右子树, 最后遍历根结点, 若二叉树为空则结束返回。