

杭州师范大学第四届程序设计新生赛

计算机科技协会

December, 12, 2021

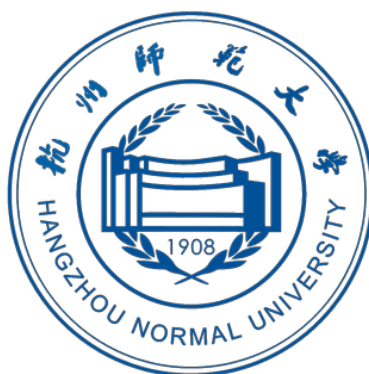


杭州师范大学
HANGZHOU NORMAL UNIVERSITY

Problems

A	解开束缚缠丝 II
B	7 的意志
C	迷宫的十字路口
D	转动命运之轮
E	计算最小值
F	月光奏鸣曲
G	电子表校对
H	简单的 LRU 问题
I	好想听肆宝唱歌啊
J	毁灭凤凰人
K	欢迎来到杭师大
L	Ayanoto 变形记
M	P 龙学长的教诲

Do not open before the contest has started.



Problem A. 解开束缚缠丝 II

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

在打败双胞胎外星人梅珠朗，解开束缚缠丝后，胜利队也获得了这些束缚缠丝分解后的产物。崛井队员决定利用这些分解产物进行分析。

已知分解十分彻底，所有分解产物均由单个区分大小写的英文字母表示，现在崛井队员可以选出一些产物，并对这些产物进行任意排列，构成一个长度尽可能长的回文串。

现在，请你利用程序帮助崛井队员算出他能构成的最长回文串的长度是多少。

Input

第一行包含一个正整数 $T(1 \leq T \leq 10)$ ，代表有 T 组数据。

对于每一组数据：

第一行包含一个正整数 $n(1 \leq n \leq 10^5)$ ，表示分解产物的总数量。

接下来的一行包含 n 个由空格分隔的英文字母 c_i ，表示第 i 个分解产物。

Output

对于每组数据，输出一行表示能构成的最长回文串的长度。

Example

standard input	standard output
2	1
2	3
A a	
3	
v a a	

Note

“回文串”是一个正读和反读都一样的字符串，比如“level”或者“noon”等等就是回文串。

在第一组数据中，我们可以选出产物“a”排列成“a”，长度为1。

在第二组数据中，我们可以选出所有产物排列成“ava”，长度为3。

Problem B. 7 的意志

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

刚刚过去的在冰岛举行的 S11 英雄联盟总决赛上，来自 LPL 的 EDG 战队以 3 : 2 的比分战胜了来自 LCK 的 DK 战队。



EDG 夺冠.jpg

其实 EDG 的夺冠冥冥中自有 7 意:

- 中国到冰岛(两国地理中心)的距离为 7777km
- 决赛当天为 DK 战队中单选手出生的第 7777 天
- EDG 的 AD 选手一发普通攻击的伤害为 777

小铁子在看完激烈的决赛之后也想成为 7 的意志的传承人，所以小铁子需要通过 7 的考核。

考核内容为：给出一段序列，你需要准确地计算出有多少个不同的区间的和为 7777。

小铁子最近很忙，所以他向聪明的你请求帮助，帮他成为 7 的意志的传承人。

Input

第一行包含一个正整数 $T(1 \leq T \leq 10)$ ，代表有 T 组数据。

对于每一组数据：

第一行为 n ，表示有 $n(1 \leq n \leq 10^5)$ 个整数。

第二行有 n 个整数 $a_i(1 \leq a_i \leq 5000)$ 。

Output

对于每组数据，输出一个整数，代表有多少个不同的区间和为 7777。

Example

standard input	standard output
1 4 5000 2000 777 5000	2

Problem C. 迷宫的十字路口

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

您需要模拟游戏 —— 迷宫的十字路口

给定无穷大的二维平面直角坐标系，初始时刻主角位于坐标原点，并在坐标轴上分布有 N 个物品，主角若经过或停留在某个位置会自动对该位置物品进行收集

请注意游戏前提：主角的所有移动必须在坐标轴上进行，若某次操作后，主角的最终位置离开了坐标轴，则会被强行移动到原点

您需要模拟 Q 次操作，操作仅会包含如下 4 种

- 1 X|Y step
- 2 X|Y
- 3
- 4 T

操作 1，沿着 X 或 Y 轴正方向走 $step$ 步，注意 $step$ 可能会出现负数 ($-10^5 \leq step \leq 10^5$)

操作 2，此时 X 或 Y 轴上的物品会以原点为中心对称变换位置

操作 3， X 轴上的所有物品绕原点逆时针旋转 90 度， Y 轴上的所有物品绕原点顺时针旋转 90 度

操作 4，主角会乘坐时光机回到第 T 次操作执行前所在的状态（包括坐标系的状态,主角的位置和主角获得的物品），注意仅仅会将状态变为第 T 次操作前一次的状态（假设当前操作是输入的第 i 个操作，那么执行完此次操作后，仍会进行第 $i+1$ 次操作,且保证 $1 \leq T \leq i$ ）

请输出执行完 Q 次操作后所获得的物品个数

Input

输入第一行 N 表示初始时刻有 N 个物品 ($0 \leq N \leq 2000$)

接下来 N 行 (x_i, y_i) ，表示在 (x_i, y_i) 存在一个物品

保证所有物品落在坐标轴上，且不存在 $(x_i, y_i) = (0, 0)$ ，其中 $(-10^9 \leq x_i, y_i \leq 10^9)$

接下来输入 Q 表示操作次数 ($1 \leq Q \leq 2000$)

接下来 Q 行的格式请参考题面

保证输入的所有数字都是整数

Output

输出 ans 表示最终得到的物品数

Example

standard input	standard output
5 0 1 0 2 -1 0 2 0 100 0 8 1 X 2 1 Y 5 1 Y 0 3 2 Y 1 X 2 4 1 4 3	1

Note

对于样例 1:

初始时刻

主角位置 $(0,0)$ ，物品数 0。

经过操作 1，主角得到 $(2,0)$ 的物品，此时得到物品数 1。

经过操作 2，主角走到 $(2,5)$ 此时不在坐标轴上，于是回到 $(0,0)$ 得到物品数 1。

经过操作 3，主角走到 $(0,0)$ 物品数 1，此时物品在 $(1,0)$ ， $(2,0)$ ， $(0,-1)$ ， $(0,100)$ 。

经过操作 2，Y 轴上的物品关于原点做对称，此时还剩余的物品在 $(1,0)$ ， $(2,0)$ ， $(0,1)$ ， $(0,-100)$ 。

经过操作 1，主角走到 $(2,0)$ ，此时经过 $(1,0)$ 和 $(2,0)$ 得到的物品数变为 3。

经过操作 4，主角回到操作 1 之前的状态，此时主角位置 $(0,0)$ ，得到物品数 0，物品位置都是初始时刻位置。

经过操作 5，主角回到第 3 次操作之前的状态，此时主角位置 $(0,0)$ 得到物品数 1。

输出答案 1。

Problem D. 转动命运之轮

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

n 个小朋友站成一排，他们都有一个属于他们的玩具，每个小朋友具有开心值 h_i 。

对于任意玩具都带有唯一的标号， i 号小朋友的玩具标号为 i ，不幸的是，现在这些玩具被随机分配给了这 n 个小朋友。

对于某种分配下，小朋友 i 会首先观察拿到的玩具的标号，记为 w_i 。

下面会进行若干轮操作，每轮操作后小朋友 i 会观察此时他的玩具标号 p_i ，并将这个玩具给小朋友 w_i 。

经过此轮操作后，若发现某个小朋友 i 的玩具恰好等于初始拿到的玩具标号，即 $p_i = w_i$ ，则该小朋友不再进行接下来的所有操作。

定义总开心值为每个小朋友的开心值乘一共经历的轮数的和，形式化表达：

$$H = \sum_{i=1}^n h_i \times c_i$$

其中 H 表示总开心值， c_i 表示经历的操作轮数。

现在定义 U_n 为长度 n 的排列的全集，你需要输出：

$$\sum_{P \in U_n} H_P$$

Input

输入第一行 n 表示排列的长度 ($1 \leq n \leq 2000$)。

接下来 n 个数，其中第 i 个数表示 i 号小朋友的开心值 h_i ($1 \leq h_i \leq 10^9$)。

Output

输出一个数字 ans 表示答案，由于答案可能会很大，您只需要输出 ans 取模 998244353 后的结果。

Examples

standard input	standard output
1 2	2
3 1 1 1	36
5 11 4 5 1 4	9000

Note

当 $n = 3$, 开心值 1, 1, 1, 排列为 $\{3, 1, 2\}$ 时, 小朋友会观察他们拿到的玩具标号, 有 $w_1 = 3, w_2 = 1, w_3 = 2$
第一轮操作后, 再次观察标号, 有 $p_1 = 1, p_2 = 2, p_3 = 3$

第二轮操作后,再次观察标号,有 $p_1 = 2, p_2 = 3, p_3 = 1$

第三轮操作后,再次观察标号,有 $p_1 = 3, p_2 = 1, p_3 = 2$, 发现此时满足 $w_1 = p_1, w_2 = p_2, w_3 = p_3$, 三个小朋友不再进行操作,三人经历的轮数都是 3, 可以算出 $H = 1 \cdot 3 + 1 \cdot 3 + 1 \cdot 3 = 9$

类似可以得到排列为 $\{1, 2, 3\}, \{1, 3, 2\}, \{2, 1, 3\}, \{2, 3, 1\}, \{3, 1, 2\}$ 时的 H , 对他们求和就可以得到 $ans = 36$

对于取模我们通常有如下结论

$$(x + y) \bmod M = (x \bmod M + y \bmod M) \bmod M$$

$$(x \cdot y) \bmod M = ((x \bmod M) \cdot (y \bmod M)) \bmod M$$

符号 \sum 表示求和, 如可以用 $\sum_{i=1}^{100} i$ 表示 1 到 100 的求和, 即:

$$1 + 2 + 3 + \cdots + 100$$

在本题中, 可以理解为对所有满足 $P \in U_n$ 的集合 P 进行求和。

Problem E. 计算最小值

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

给定 n 个数组，每个数组 m 个数字。

分别从每个数组中取一个数字，组成一个集合，进行如下操作：

1. 从集合中任取一个数，记作 a 。
2. 从集合中任取一个数，记作 b ，将 $|a - b|$ 计入答案。
3. 丢弃 a ，保留 b ，并将其作为新的 a ，重复步骤 2，直到数组中只剩一个数。

计算所有第 2 步中计入答案的数的和可能达到的最小值。

Input

首行给出 $n, m (2 \leq n \leq 10, 0 \leq m \leq 10^6, n \cdot m \leq 10^6)$ 表示共有 n 个数组，每个数组 m 个元素。

接下去 n 行，每行 m 个数表示数组各个元素 (a_{ij} 表示第 i 个数组第 j 个数字， $\forall i, j, 0 \leq a_{ij} \leq 10^9$)。

Output

一个整数，表示答案的最小值。

Example

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

Problem F. 月光奏鸣曲

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

给定两个大小为 $N \cdot N$ 的正方形，其中共有 $N \cdot N$ 个格子，每个格子有用数字来标识的颜色。

你可以对第一个正方形进行旋转操作，每次可以旋转绕中心顺时针旋转 90 度，也可以逆时针旋转 90 度。

求最小的操作次数，使得第一个正方形和第二个正方形匹配。

如果无法通过操作使得两个正方形匹配，请输出 -1 。

我们定义匹配为：对于任意 $1 \leq i, j \leq n$ ，有第一个正方形 (i, j) 的颜色和第二个正方形上 (i, j) 颜色相同。

Input

输入第一行 T 表示一共有 T 组测试数据 ($1 \leq T \leq 100$)。

接下来一行输入一个正整数 N 表示正方形的大小 ($1 \leq N \leq 20$)。

接下来 $2 \cdot N$ 行，每行 N 个数：

其中前 N 行表示第一个正方形的颜色。

后 N 行表示第二个正方形的颜色，其中颜色用数字标号 $color_{i,j}$ 表示 ($1 \leq color_{i,j} \leq 10^9$)。

Output

输出 T 行，每行输出 ans 表示最小的操作次数，若不能通过操作使得两个正方形匹配，输出 -1 。

Example

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

Problem G. 电子表校对

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

现在你有一个电子表，但是他的时间可能是不准确的，请你对照准确时间确定电子表的时间相差了多少。

Input

单组数据输入。

输入一共 6 行，每行为一个长度为 18 的字符串。

前三行代表实际时间，后三行代表你的电子表时间，时间为 xx 时 xx 分 xx 秒。

每个数字占一个 3×3 的大小。

0 ~ 9 的数字格式参照样例 2。

Output

输出一共 4 行。

第 1 行输出两者时间关系：

- 如果你的电子表时间比实际时间慢输出 “late”（没有引号）。
- 如果比实际时间快输出 “early”（没有引号）。
- 如果完全一样输出 “gang gang hao”（没有引号）。

第 2 行到第 4 行输出 3 行，代表两者时间差。

Examples

standard input	standard output
<pre> - - - - - _ _ _ _ _ _ _ _ _ - - - - - _ _ _ _ _ _ _ _ _ </pre>	<pre> late - - - - - _ _ _ _ _ </pre>
<pre> - - - - - _ _ _ _ _ _ _ _ _ _ - - - - - _ _ _ _ _ _ _ _ _ _ </pre>	<pre> early - - - _ _ _ _ _ </pre>

Problem H. 简单的 LRU 问题

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

众所周知，电脑能够运行远大于其内存的程序，这得益于**局部性原理**，也就是说你现在用到的内存及其周围的内存空间，在不久的将来也会被频繁的调用，想想你写的 **for** 语句就能明白。

因此操作系统会动态的将暂时用不到的内存块复制转移到了外存中，同时将正在或不久的将来会用到的部分留在了内存中。物理上内存的总量并没有增加，但逻辑上你能使用的内存变多了。

现在我们简单的模拟一下操作系统的工作。你有一大块内存空间，它被平均地分为了 n 块，硬盘中也有大量的空间，它们被分为了若干和内存中块大小一样的块。当操作系统想要访问某一块空间时，它会查看内存中时候存在这一块，如果有就直接访问，如果没有，会去外存中寻找找到它，并将它放入内存。如果此时内存已满，就不得不将内存中现有的块替换出去。

你正在使用一个极其消耗内存的程序，频繁的访问了一系列内存空间。在此我们将其简化为一个数字序列 a ，表示你依次访问了这些内存空间。

操作系统采用了 **最近最久未使用算法(LRU)**，也就是选择最近最长时间没有使用的块予以淘汰。注意这一步只发生在有新的块想从外存进入内存。

聪明的你想必已经理解了这一切，请你画一张表格描述这一切叭。

如果你没搞懂什么是 **LRU**，可以阅读以下算法：

对于每个新的「内存块」请求，执行以下操作：

- 0. 如果已经有相同的「内存块」存在于内存中，将该「内存块」置为最低优先级，原先优先级低于它的自动提升一级。
- 1. 否则如果内存中的 n 个块没有摆满，直接将新的「内存块」请求作为最低优先级放入内存中。
- 2. 否则内存中 n 个块已被摆满，将当前内存中优先级最高的块清除出内存，将其他内存块优先级自动提升一级，将新的「内存块」作为最低优先级放入内存中。

Input

第一行两个数字 $n, m (1 \leq n \leq 5, 1 \leq m \leq 100)$ 表示内存被平均分为了 n 个块，操作系统共按序访问了 m 个内存块。

第二行一个长度为 m 的序列 a ， $a_i (0 \leq a_i \leq 65535)$ 表示第 i 个被访问的内存块的序号。

Output

输出共 $m + 1$ 行 $n + 1$ 列。

第一行标记内存块淘汰的优先级，数字越小优先级越高。

接下去 m 行每行 $n + 1$ 列，第一列表示访问内存的次数，下标从 0 开始，之后 n 列按淘汰优先级列出每个内存块的编号，无编号则留空。

所有内存块编号采用 4 位 16 进制数表示，优先级和访存次数用 2 位 16 进制数表示。

具体格式参考样例。

Examples

standard input	standard output
3 4 0 1 2 1	+-----+-----+-----+-----+ 0x00 0x01 0x02 +-----+-----+-----+-----+ 0x00 0x0000 +-----+-----+-----+-----+ 0x01 0x0000 0x0001 +-----+-----+-----+-----+ 0x02 0x0000 0x0001 0x0002 +-----+-----+-----+-----+ 0x03 0x0000 0x0002 0x0001 +-----+-----+-----+-----+
3 4 1 2 65535 255	+-----+-----+-----+-----+ 0x00 0x01 0x02 +-----+-----+-----+-----+ 0x00 0x0001 +-----+-----+-----+-----+ 0x01 0x0001 0x0002 +-----+-----+-----+-----+ 0x02 0x0001 0x0002 0xffff +-----+-----+-----+-----+ 0x03 0x0002 0xffff 0x00ff +-----+-----+-----+-----+

Problem I. 好想听肆宝唱歌啊

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

又到了下班的时间，晚安~

— 火西肆

火西肆结束一天的直播下播了。

虽然 stff577 喜欢听火西肆唱歌，但是没有听到自己特别喜爱的歌还是有些遗憾的。

火西肆直播间的点歌规则为「先到先服务」，即开播时观众最先点的歌会加入队列中，也可以使用钞能力。

但是作为没有经济收入的大学生的 stff577 显然不会使用钞能力。

他知道火西肆接下来马上就要开播了，所以他决定相信自己多年单身的手速，点到自己特别喜爱的歌。

stff577 有 n 首特别喜爱的歌，他对每首歌的喜爱程度都能用一个整数来表示，且喜爱程度各不相同，喜爱程度越大，就说明他越想点这首歌。

由于 stff577 没有错过火西肆的每场直播，所以他知道他最想点的前 k 首歌都已经被人点过了，显然他不想浪费点歌的机会，所以他会选择第 $k+1$ 首最想点的歌。

火西肆的直播马上就要开始了，而 stff577 实在太过紧张，不知道自己该点的歌是哪首，所以请你帮助他尽快找到他想点的歌。

Input

第一行包含一个数字 $n(1 \leq n \leq 10^5)$ ，代表有 n 首歌。

接下来 n 行，第 i 行包含一个正整数 $w_i(1 \leq w_i \leq 10^9)$ 和一个字符串 $s_i(1 \leq |s_i| \leq 15)$ ，代表第 i 首歌的喜爱程度和歌名。

题目保证每首歌的喜爱程度和歌名各不相同。

最后一行包含一个正整数 $k(0 \leq k < n)$ ，代表已经被点过的最想点的歌的数量。

Output

输出一行，代表想点的歌的歌名。

Examples

standard input	standard output
4 1 flos 3 Yellow 9 Starduster 1000000000 Kawakiwoameku 3	flos
1 1000000000 CryingforRain 0	CryingforRain

Problem J. 毁灭凤凰人

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

在 ygo 中有一张臭名昭著的怪兽卡「命运英雄 毁灭凤凰人」， $ATK2500$ ， $DEF2100$ ，每次毁灭凤凰人被破坏，他可以在下个回合从墓地复活（只要毁灭凤凰人被除外了，他就不能从墓地复活）。除外指移到除外区，是一个不同于场上、手牌和墓地的区域，从墓地复活的怪兽卡不能从除外区复活。

你希望能在这一回合中解掉毁灭凤凰人并且让他不能复活，你手上有 n 张卡牌，可能由以下卡片组成：

- 0 表示怪兽卡，如果对方的毁灭凤凰人是攻击表示的，那么只要这张怪兽卡的 ATK 大于等于毁灭凤凰人的 ATK 就可以将毁灭凤凰人破坏并送入墓地，如果对方的毁灭凤凰人是守备表示的，那么只要这张怪兽卡的 ATK 大于毁灭凤凰人的 DEF 就可以将毁灭凤凰人破坏并送入墓地。
- 1 表示「墓穴的指名者」，非怪兽卡，他可以将对方墓地一张怪兽卡除外（只能使墓地的怪兽卡除外）。
- 2 表示「黑核」，非怪兽卡，他可以将场上的毁灭凤凰人除外，但是必须丢弃一张其他手牌才能发动（只能除外场上的，即没有被破坏送入墓地的怪兽）。

简单来说，你要合理运用你的手牌将毁灭凤凰人除外，比如先使用怪兽卡将毁灭凤凰人破坏并送入墓地，再使用「墓穴的指名者」将毁灭凤凰人除外，或者丢弃一张手牌使用「黑核」直接将场上的毁灭凤凰人除外。

Input

第一行两个数字 n, m ($0 \leq n \leq 10, 0 \leq m \leq 1$) 表示你的手牌数量和毁灭凤凰人的表示形式，0 表示攻击表示，1 表示守备表示。

接下来 n 行每行输入一个数字 x_i ($0 \leq x_i \leq 2$)，表示第 i 张卡牌的种类，如果是 0，则会再输入一个数字 k , ($0 \leq k \leq 10^4$) 表示攻击力。

Output

如果可以解掉毁灭凤凰人并使他不能复活，就在一行内输出 “haoye”(不包括引号)，否则输出 “QAQ”(不包括引号)。

Example

standard input	standard output
2 0 0 2500 1	haoye

Problem K. 欢迎来到杭师大

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

我最亲爱的王小美:

昨夜我辗转反侧在床上睡不着, 听着窗外淅淅沥沥的雨声, 我想起来我和你曾经在一起的时光, 尽管我们已经不再一起了, 但是我还是很感谢你出现在我的生命中, 我想到我曾经牵着你的手漫步在校园内, 湖面映着夜晚的廊桥, 星星在湖中跳舞, 那天的星光真的很美, 但是看到你的眼睛之后我知道。以后看星星只需要看你就好了。

虽然我们现在已经分开, 但是我知道当时那一刻砰然的心动我将永远铭记, 并且那一刻现在还时时让我心动。我不知道将来我是否还会遇见你, 但我冥冥之中仍然坚信我们会在某个时空永远的在一起, 我们会在春天花野中轻嗅花香, 看蝴蝶在花间曼舞; 会在夏天坐在树下倾听蝉鸣; 也会在秋天的稻田中享受那一缕属于我们之间的麦香……

很怀念我们之前在一起的时光, 那时的我们在爱中成长, 互相学习。我曾经也很期待我们将来的时光, 那时的我们肯定还会依偎在一起, 成为世界上最幸福的人吧……

正如《The Notebook》所说的最好的爱是那种唤醒灵魂, 让我们追求更多的爱, 是种在我们心中的火, 让我们的心灵平静。这就是你给我的。这也是我希望给你的。

我爱你, 再见。

爱你的俊俊
2021 年 9 月 12 日

纵然时间在流逝, 但总有一个人我们无法忘怀。

俊俊理了理自己的刘海, 这时的俊俊已经考上了大学, 他走入了杭州师范大学的校园。

校园的美景固然美丽, 但是并没有吸引住俊俊的眼光, 忽然一道倩影映入了俊俊的眼帘, 柔顺的蓝色短发和轻巧可爱的脸庞, 还有那散发着光芒的眼睛, 俊俊突然呆住了。

「这……这不是王小美吗?!!!!」俊俊一个箭步走了上去, 王小美的目光也被俊俊吸引了, 那曾经带给她无数快乐回忆的、也曾让她魂牵梦绕的男孩, 现在如今出现在她的面前……

「纵然世间有千百万朵艳丽的鲜花, 我也只钟情我那一朵」, 俊俊说道。

王小美难以掩饰自己的心情, 眼泪从脸颊滑下, 她笑着对俊俊说「欢迎来到杭州师范大学」。

俊俊真的很爱王小美但是他在梦里都未曾想过他会再次遇见她, 激动的俊俊在大脑中重复了很多遍王小美的话……

Input

数字 $n(1 \leq n \leq 100)$ 。

Output

n 遍 “Welcome to HZNU” (不含引号), 每句话一行。

Example

standard input	standard output
1	Welcome to HZNU

Problem L. Ayanoto 变形记

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

可怜的 Ayanoto 被巫师变成了一只小青蛙。

巫师把他放到圆环上的某一点，并告诉他，如果你从这里出发，并能再次到这里，就能变回人类。

圆环上有 n 个点，简便起见编号为 0 到 $n-1$ 。

由于是青蛙的体型，他每一步只能跳确切的 x 个长度。

比如说，如果他现在在 0 号点，那么下一步，他可以向前跳到第 x 个点。

请你判断，经过不断的跳跃后，Ayanoto 是否有可能变为人类。

Input

t 组输入 ($1 \leq t \leq 10^3$)。

每组输入包含两个数字 $n(2 \leq n \leq 10^6), x(0 \leq x \leq 10^6)$ 。

Output

t 行输出，如果 Ayanoto 可以变为人类输出 “yes”（没有引号），否则输出 “no”（没有引号）。

Examples

standard input	standard output
2	yes
3 1	yes
4 2	
2	yes
3 2	no
13 0	

Problem M. P 龙学长的教诲

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

有一天，小包包遇到一个很棘手的问题，问遍了周围的同学都不能解释。

在小包包遇到人生挫折的时候想起了一位学长——P龙学长（曾经叱咤 HZNU-ACM 实验室的传奇人物）。

但 P 龙学长 有一个小毛病，说话东一句西一句。

这个毛病形式上的说，一句话有 n 个单词，每个单词为 $a_i (1 \leq i \leq n)$ 。

P 龙学长会以 “ $a_1 \ a_3 \ a_5 \cdots a_6 \ a_4 \ a_2$ ” 的形式说。

实际上他想说的是：

$$a_1 \ a_2 \ a_3 \ a_4 \ a_5 \ a_6 \cdots$$

例如 P 龙学长说：“This an problem easy is.”

但他其实是想说：“This is an easy problem.”

小包包非常想知道 P 龙学长 的教诲，请你帮帮他吧！

Input

第一行有一个数字 $T (1 \leq T \leq 10^2)$ ，代表接下去有 T 行。

然后每行都有若干个单词组成，每个单词有且仅有小写字母 ‘a’ 到 ‘z’、大写字母 ‘A’ 到 ‘Z’ 和数字 $0 \sim 9$ 。

值得注意的是：每行教诲最后都会紧跟一位标点符号（只会出现 ‘.’、‘!’ 或 ‘?’ 三个字符中的一个）。

每个单词长度 $1 \leq len \leq 30$ ，每行的单词数不超过 1000 个。题目保证每个单词之间只存在一个空格符。

Output

有 T 行，每行输出 P 龙学长 通顺的语句吧！

每行结尾最后的特殊字符依旧在最后！

Example

standard input	standard output
2 This an problem easy is. a c e f d b!	This is an easy problem. a b c d e f!