

第十九届吉林省大学生 程序设计竞赛

正式赛

2026 年 5 月 23 日



试题列表

A	Manhattan Cycle
B	Hello, 72!
C	Maple
D	隐藏的 k 元组
E	洛克王国世界
F	计树
G	题列序 1
H	图
I	古明地恋世界第一
J	翻转逆序对
K	拆分树
L	顺子
M	水晶城堡

本试题册共 13 题，16 页。

如果您的试题册缺少页面，请立即通知志愿者。

题目 A. Manhattan Cycle

小 G 是生活在 **H** 星球的智慧生命，在这个星球上，存在着无边无际的花丛，小 G 十分喜欢在花丛中跳跃。在花丛中玩耍的第 $10^9 + 7$ 天，小 G 学会了什么是曼哈顿距离，并立刻用它创作了一个游戏。

小 G 的游戏是这样的，选取一片正方形的花丛，然后将其分为 n^2 个完全相同的小正方形，排成每行每列都是 n 个的样子，然后在所有正方形的顶点之间跳跃，每次跳跃时，起点和终点的距离都必须是一个给定的正整数 K 。而且，小 G 不希望找不到自己的家，所以小 G 希望经过若干次跳跃之后，最终回到起点。

小 G 立刻找出了一个简单的方案，具体来说，首先小 G 选择位于角落的点作为起点，然后选择一个和该点的曼哈顿距离为 K 的点，跳跃过去，然后再跳跃回来，小 G 通过这个方法立刻找出了所有跳跃次数为偶数的方案。但是，小 G 认为这太简单了，他想要找到一些更加强大的方案，所以，他向你求助，希望你能帮他找出一些跳跃次数为奇数的方案。

根据前面的经验，小 G 认为只需要找出跳跃次数最少的方案，就能构造跳跃次数更多的方案。所以，你只需要给出所有跳跃次数为奇数的方案中，所用的跳跃次数的最小值是多少，或者回答不存在这样的方案。而且，由于小 G 还没有想好选择将花丛划分为多少份，也没有确定跳跃的距离，所以他会询问你 T 次，每次给出正整数 n, K ，你需要依次回答他的所有询问。

为了让作为人类的你更好地理解 **H** 星智慧生命的神秘想法，我们可以把花丛划分后所有正方形的顶点看作是平面直角坐标系中所有横纵坐标均为整数，且在 $[0, n]$ 之中的点，我们称这样的点为特殊点。在两个特殊点 $\mathbf{I}(x_1, y_1), \mathbf{T}(x_2, y_2)$ 之间可以跳跃的条件为 $|x_1 - x_2| + |y_1 - y_2| = K$ 。

对于一次给定正整数 n, K 的询问，小 G 希望得到的答案如下：是否存在从某个特殊点出发，按照上述规则在特殊点之间跳跃，最终回到出发点，并且跳跃次数为奇数的方案，如果存在，给出所有这样的方案中跳跃次数的最小值，否则，输出 -1 。

输入

第一行有一个整数 T ($1 \leq T \leq 10^5$)，表示数据组数。接下来 T 行，每行描述一组数据。

每组数据包含两个整数 n 和 K ($1 \leq n \leq 10^8, 1 \leq K \leq 2n$)。

输出

对于每组数据，输出一行一个整数。如果存在步数为奇数的跳跃方案，则表示所有这样的方案中跳跃次数的最小值；否则输出 -1 ，表示无解。

样例

standard input	standard output
3	-1
1 1	3
2 2	-1
4 8	

题目 B. Hello, 72!

tarjen 最近在学习日语。在日语中，数字 1 到 9 的读法如下：

数字	日文读法
1	ichi
2	ni
3	san
4	yon
5	go
6	roku
7	nana
8	hachi
9	kyuu

要读一个多位数，只需从左到右依次拼接每个数字的日文读法。例如，15 读作 *ichigo*，893 读作 *hachikyuusan*。

tarjen 最喜欢的数字是 72，他现在想考考你，72 的日文读法是什么？

输入

本题没有输入。

输出

输出一行一个字符串作为答案。

题目 C. Map1e

给定一个正整数 N ，定义全 1 数 $R(k)$ 为 k 个 1 组成的数，即 $R(k) = \underbrace{111 \dots 1}_{k \text{ 个}}$ 。

请找到最大的正整数 k ，使得 $R(k)$ 是 N 的因子，并输出 k 。

注意 $R(1) = 1$ 是所有正整数的因子，因此答案至少为 1。

输入

第一行有一个整数 T ($1 \leq T \leq 5 \times 10^5$)，表示数据组数。接下来 T 段，每段描述一组数据：

- 第一行包含一个正整数 N ($1 \leq |N| \leq 10^5$ ，其中 $|N|$ 表示 N 的十进制位数；保证 N 没有前导零)。

数据保证 $\sum |N| \leq 5 \times 10^5$ 。

输出

对于每组数据，输出一行一个正整数 k 。

样例

standard input	standard output
3	3
1221	2
99	1
7	

注释

在第一个样例中， $1221 = 111 \times 11$ ，所以 $R(3) = 111$ 是 N 的因子。 $R(4) = 1111$ 不是 N 的因子，因此答案为 3。

在第二个样例中， $99 = 11 \times 9$ ，所以 $R(2) = 11$ 是 N 的因子。 $R(3) = 111 > 99$ ，因此答案为 2。

在第三个样例中，7 不是 11 的倍数，因此答案为 1。

题目 D. 隐藏的 k 元组

这是一个交互题。

有一个隐藏的划分，将整数 $1, 2, \dots, n$ 分成 $\frac{n}{k}$ 个互不相交的 k 元组。保证 n 是 k 的倍数。你需要通过询问找出所有隐藏的 k 元组。

一次询问中，你可以选择一个集合 $S \subseteq \{1, 2, \dots, n\}$ ，交互库会返回一个整数，表示有多少个隐藏的 k 元组被完整包含在 S 中。

你的询问次数不能超过 $n \times \lceil \log_2 n \rceil$ 次。

术语说明： $\lceil \cdot \rceil$ 是上取整符号， $\lceil x \rceil$ 表示不小于 x 的最小整数。例如 $\lceil 7 \rceil = 7$ ， $\lceil 3.14 \rceil = 4$ 。

输入

程序开始时，交互库会输出一行两个整数 n 和 k ($2 \leq n \leq 300$, $2 \leq k \leq n$, 并且 n 是 k 的倍数)。

隐藏的划分由交互库保存，不会直接给出。

在你每次输出一行合法询问之后，交互库会返回一行一个整数 r ，表示有多少个隐藏的 k 元组被完整包含在你询问的集合中。

输出

你可以进行如下形式的询问：

$$? \ c \ x_1 \ x_2 \ \dots \ x_c$$

其中 $0 \leq c \leq n$ ，且 $x_1, x_2, \dots, x_c$ 必须是两两不同的整数，满足 $1 \leq x_i \leq n$ 。

该询问表示你选择集合 $S = \{x_1, x_2, \dots, x_c\}$ 。交互库会返回一个整数 r ，表示有多少个隐藏的 k 元组被完整包含在 S 中。

当你确定答案后，需要输出：

$$! \ a_1 \ a_2 \ \dots \ a_n$$

其中 a_i 表示元素 i 所在的元组编号。编号必须满足 $1 \leq a_i \leq \frac{n}{k}$ 。如果两个元素属于同一个隐藏元组，则它们的编号必须相同；如果两个元素属于不同隐藏元组，则它们的编号必须不同。元组编号的顺序可以任意。

你的询问次数不能超过 $n \times \lceil \log_2 n \rceil$ 次。输出最终答案后，你的程序应立即结束。

注意，每次输出询问或最终答案后都必须刷新输出缓冲区。例如，在 C++ 中可以使用 `fflush(stdout)` 或 `cout << flush`。

如果你的输出格式非法、询问次数超过限制，或最终答案错误，将得到 Wrong Answer 或 Presentation Error。

交互库是非自适应的，即所有的 k 元组在交互前就已经确定好，不会随着询问的发生而改变。

样例

standard input	standard output
6 2	? 3 1 3 5
1	? 4 2 3 4 5
1	? 6 1 2 3 4 5 6
3	! 1 2 1 3 3 2

注释

在下面的例子中， $n = 6$ ， $k = 2$ ，隐藏的元组为 $\{1,3\}$ 、 $\{2,6\}$ 、 $\{4,5\}$ 。

程序	交互库
	6 2
? 3 1 3 5	1
? 4 2 3 4 5	1
? 6 1 2 3 4 5 6	3
! 1 2 1 3 3 2	

- 询问 $\{1,3,5\}$ ：元组 $\{1,3\} \subseteq S$ ，答案为 1。
- 询问 $\{2,3,4,5\}$ ：元组 $\{4,5\} \subseteq S$ ，答案为 1。
- 询问 $\{1,2,3,4,5,6\}$ ：三个元组都被包含，答案为 3。
- 输出 `! 1 2 1 3 3 2` 表示：元素 1,3 组成第 1 组；元素 2,6 组成第 2 组；元素 4,5 组成第 3 组。

题目 E. 洛克王国世界

tarjen 最近在玩洛克王国，他想要捉一些异色精灵。

在刷异色精灵的过程中，*tarjen* 需要先不断捕捉普通精灵，捕捉普通精灵可以积累异色池的污染进度。当污染进度达到要求时，地图上会刷新出对应池子的污染精灵。在击败污染精灵后，有机会遇到异色精灵。

不过，地图上的污染精灵数量有限。如果污染精灵刷得太多，旧的污染精灵会被新的污染精灵挤掉。

同时，游戏中存在刷新异色精灵的保底机制。每个异色池有独立的保底计数。每击败一次当前池子的污染精灵，该池子的保底计数加一。当某个池子的保底计数达到 80 时，本次击败污染精灵后一定会遇到异色精灵。

具体规则如下。

共有 n 种普通精灵， A 个家族和 B 个属性。第 i 种普通精灵属于家族 fa_i ，属性为 el_i 。

游戏中共有两类异色池：

- 家族池 F_x ：第 x 个家族的异色池。
- 属性池 E_x ：第 x 个属性的异色池。

每个异色池 P 维护了四个值：

- $progress_P$ ：污染进度，初始为 0。
- $pity_P$ ：保底计数，初始为 0。
- $need_P$ ：刷新一只污染精灵所需的普通捕获次数（给定常数）。
- $luck_P$ ：幸运阈值（给定常数）。

地图中最多同时存在 m 只污染精灵。所有污染精灵按刷新时间从早到晚排成一个队列。若数量超过 m ，则不断移除队首的污染精灵，直到数量不超过 m 。被地图容量挤掉的污染精灵不会被视为击败，也不会影响任何池子的保底计数。

共有 q 次操作，每次操作为以下两种之一：

1. 普通捕获

C T x c

表示 *tarjen* 连续捕获第 x 种普通精灵 c 次，并将这 c 次捕获计入某一类异色池。若 $T = F$ ，则计入家族池 F_{fa_x} ；若 $T = E$ ，则计入属性池 E_{el_x} 。

设本次捕获计入的池子为 P ，则令 $progress_P += c$ 。然后，每当 $progress_P \geq need_P$ ，就刷新出一只属于池子 P 的污染精灵，并令 $progress_P -= need_P$ 。

注意，一次普通捕获可能会刷新出多只污染精灵。

将本次刷新出的污染精灵依次加入队列尾部，若数量超过 m 则移除队首的污染精灵。

2. 击败污染精灵

B k r

表示 *tarjen* 选择当前地图中从旧到新第 k 只污染精灵进行挑战。

若当前地图中的污染精灵数量不足 k ，则输出 MISS，本次操作不改变任何状态。

否则，设该污染精灵属于池子 P 。将其从地图中移除，并令 $pity_P += 1$ 。

然后进行异色判定。每个池子 P 有一个幸运阈值 $luck_P$ 。若满足 $r \leq luck_P$ 或 $pity_P = 80$ ，则本次

遇到异色，并令 $\text{pity}_P = 0$ ；否则不重置保底计数。具体输出格式见输出部分。

输入

第一行包含五个整数 n, A, B, m, q ($1 \leq n, A, B, m, q \leq 2 \times 10^5$)，分别表示普通精灵种类数、家族数量、属性数量、地图中最多同时存在的污染精灵数量以及操作次数。

接下来 n 行，每行两个整数 fa_i, el_i ($1 \leq fa_i \leq A, 1 \leq el_i \leq B$)。

接下来一行 A 个整数 $\text{need}_{F_1}, \dots, \text{need}_{F_A}$ ($1 \leq \text{need}_{F_i} \leq 10^9$)。

接下来一行 B 个整数 $\text{need}_{E_1}, \dots, \text{need}_{E_B}$ ($1 \leq \text{need}_{E_i} \leq 10^9$)。

接下来一行 A 个整数 $\text{luck}_{F_1}, \dots, \text{luck}_{F_A}$ ($0 \leq \text{luck}_{F_i} \leq 10^9$)。

接下来一行 B 个整数 $\text{luck}_{E_1}, \dots, \text{luck}_{E_B}$ ($0 \leq \text{luck}_{E_i} \leq 10^9$)。

接下来 q 行，每行描述一个操作。格式为 C T x c ($T \in \{F, E\}, 1 \leq x \leq n, 1 \leq c \leq 10^{18}$) 或 B k r ($1 \leq k \leq 10^{18}, 1 \leq r \leq 10^9$)。

输出

对于每个 B 操作，输出一行结果：

- 若地图中不足 k 只污染精灵，输出 MISS；
- 否则，若遇到异色，输出 SHINY F x 或 SHINY E x ；
- 否则，若未遇到异色，输出 NORMAL F x p 或 NORMAL E x p ，其中 x 为对应异色池编号， p 为该池子当前的保底计数。

样例

standard input	standard output
5 2 2 5 10	NORMAL E 2 1
1 2	NORMAL E 1 1
1 1	MISS
2 2	NORMAL E 2 2
1 1	NORMAL F 1 1
2 2	MISS
2 2	
2 3	
40 5	
4 41	
C E 5 9	
C E 4 5	
B 2 608984673	
B 4 918347739	
B 4 37535012	
C E 5 7	
B 1 712704464	
C F 4 7	
B 3 270899745	
B 5 363752455	
3 2 1 3 8	NORMAL E 1 1
1 1	SHINY F 2
1 1	MISS
2 1	NORMAL F 1 1
2 3	
2	
0 5	
10	
C F 1 2	
C E 2 4	
B 2 50	
C F 3 6	
B 3 3	
B 5 1	
C F 1 2	
B 3 100	

注释

对于样例 2，我们用 $F1$ 表示池子 $F1$ 的污染精灵， $E1$ 表示池子 $E1$ 的污染精灵。

- 前两次捕获后，地图中的污染精灵为 $[F1, E1, E1]$ 。
- 执行 B 2 50：击败第 2 只 ($E1$)。 $50 > \text{luck}_{E1} = 10$ 且 $\text{pity}_{E1} = 1 \neq 80$ ，输出 NORMAL E 1 1。
- 之后捕获第 3 种精灵并计入家族池，刷新 2 只 $F2$ 。地图容量为 3，旧的被挤掉后变为 $[E1, F2, F2]$ 。
- 执行 B 3 3：击败第 3 只 ($F2$)。 $3 \leq \text{luck}_{F2} = 5$ ，遇到异色，输出 SHINY F 2。
- 执行 B 5 1：当前地图不足 5 只，输出 MISS。
- 最后一次击败的是 $F1$ ，未遇到异色，输出 NORMAL F 1 1。

题目 F. 计树

给出一个整数 K ，以及两个长度为 $2^K - 1$ 的数组 $a_1, a_2, \dots, a_{2^K-1}$ 和 $b_1, b_2, \dots, b_{2^K-1}$ 。
有一个大小为 2^K 的无向完全图 G ，点编号范围为 $0 \sim 2^K - 1$ 。

对于图 G 的一棵生成树 T ，定义

$$\begin{aligned} A(T) &= \prod_{(u,v) \in T} a_{u \oplus v}, \\ B(T) &= \sum_{(u,v) \in T} b_{u \oplus v}, \\ C(T) &= \bigoplus_{(u,v) \in T} (u \oplus v). \end{aligned}$$

你需要对于每个 $0 \leq x < 2^K$ ，求出

$$\left(\sum_{\substack{T \\ C(T)=x}} A(T) B(T)^p \right) \bmod 998244353,$$

其中 p 为一个给定的常数。

规定 $0^0 = 1$ 。

术语说明：无向完全图表示任意两个不同点之间都有一条无向边；生成树表示选出若干条边，使得所有点连通且不存在环；符号 $\oplus$ 表示按位异或。

输入

第一行包含两个整数 K, p ($1 \leq K \leq 16, 0 \leq p \leq 5$)。
接下来一行包含 $2^K - 1$ 个整数，第 i 个整数表示 a_i ($0 \leq a_i < 998244353$)。
接下来一行包含 $2^K - 1$ 个整数，第 i 个整数表示 b_i ($0 \leq b_i < 998244353$)。

输出

一行包含 2^K 个整数，第 i 个整数表示 $x = i - 1$ 时的答案。

样例

standard input	standard output
2 0 1 1 1 1 2 3	4 4 4 4
2 2 2 3 5 1 4 2	5880 5416 10464 9640

题目 G. 题列序 1

给定一个长度为 n 且值域在 $[0, 2]$ 的整数序列 a 。定义一次操作如下：

1. 选择下标 x ($1 \leq x \leq n - 2$);
2. 令 $s = (a_x + a_{x+1} + a_{x+2}) \bmod 3$ ，然后将 a_x, a_{x+1}, a_{x+2} 同时变为 s 。

你需要做若干次操作（可以不做）以最大化 a 序列所有数的和。同时，你需要给出一种合法操作方案，该方案的操作次数不超过 $\left\lfloor \frac{5}{7}n \right\rfloor + 100$ 。可以证明总是存在满足操作次数限制的方案。

输入

第一行有一个整数 T ($1 \leq T \leq 1000$)，表示数据组数。接下来 T 段，每段描述一组数据：

- 第一行输入一个整数 n ($3 \leq n \leq 10^6$)，表示序列 a 的长度。
- 第二行包含 n 个整数，第 i 个整数表示 a_i ($0 \leq a_i \leq 2$)。

数据保证 $\sum n \leq 10^6$ 。

输出

对于每组数据：

- 第一行输出两个整数 s 和 K ，分别表示序列和的最大值以及你给出的方案的操作次数，你需要保证 $0 \leq K \leq \left\lfloor \frac{5}{7}n \right\rfloor + 100$ 。
- 接下来一行输出 K 个正整数，第 i 个整数表示第 i 次操作选择的 x 。

样例

standard input	standard output
3	3 1
3	1
0 1 0	10 4
5	2 2 3 1
0 2 1 2 2	14 5
7	1 3 4 5 1
1 1 1 1 1 1 1	

题目 H. 图

给定一个 n 个点 m 条边的简单无向连通图 G 。你需要选择两条不同的边删去，得到新图 G' ，要求 G' 仍然连通。

令删掉的两条边为 (p, q) 和 (u, v) 。你需要最小化 p 到 q 在 G' 上的最短路长度与 u 到 v 在 G' 上的最短路长度之和，输出这个最小值，并求出有多少种无序删边方案可以达到这个最小值。

输入

第一行有一个整数 T ($1 \leq T \leq 5000$)，表示数据组数。接下来 T 段，每段描述一组数据。对于每组数据：

- 第一行包含两个整数 n, m ($4 \leq n \leq 5000, 5 \leq m \leq 5000$)。
- 接下来 m 行，每行包含两个整数 u, v ，表示一条边。

数据保证 $\sum n, \sum m \leq 5000$ 。给出的图无重边、无自环，且至少存在一种删边方案使得图仍然连通。

输出

对于每组数据，输出一行包含两个整数，分别表示最小值和达到最小值的无序删边方案数。

样例

standard input	standard output
4	4 15
4 6	4 4
1 2	6 12
1 3	4 4
1 4	
2 3	
2 4	
3 4	
4 5	
1 2	
2 3	
3 4	
1 4	
1 3	
5 6	
1 3	
2 3	
1 4	
2 4	
1 5	
2 5	
5 7	
1 2	
2 3	
3 4	
4 5	
1 5	
2 5	
1 4	

题目 1. 古明地恋世界第一

Koishi 在路上捡到了一个长度为 n 的正整数序列 $a = [a_1, a_2, \dots, a_n]$, 满足 $1 \leq a_i \leq n$ 。现在 Koishi 计算出了 a 序列的前缀最大值序列 $b = [b_1, b_2, \dots, b_n]$, 其中 b_i 表示 a 序列前 i 项的最大值, 即

$$b_i = \max\{a_1, a_2, \dots, a_i\} = \max_{1 \leq j \leq i} \{a_j\}$$

在得到两个序列后, Koishi 将它们混合到了一起, 即定义长度为 $2n$ 的序列 $c = [a_1, a_2, \dots, a_n, b_1, b_2, \dots, b_n]$, 并将 c 从小到大排序。

由于 Koishi 是无意识少女, 她丢掉了 a 和 b 的值, 只剩下了 c 序列。

由于 Koishi 是无意识少女, 她将 c 序列给了你——二十一世纪最可爱的大学生, 希望你找到任意一个合法的 a 序列。可能有多个合法的 a 序列, 这时你只需要求出任意一个即可; 也可能没有任何一个合法的 a 序列, 此时你需要报告无解。

输入

第一行有一个整数 T , 表示数据组数。接下来 T 段, 每段描述一组数据。对于每组数据:

- 第一行输入一个整数 n 表示序列 a 的长度;
- 第二行输入一个长度为 $2 \times n$ 的序列 c , 相邻整数由一个空格隔开, 保证 c 序列单调不减, 且 $1 \leq c_i \leq n$ 。

数据保证 $1 \leq \sum n \leq 3 \times 10^5$ 。

输出

对于每组数据:

- 若有解, 第一行输出一个字符串 “Yes”, 第二行输出 n 个整数 $[a_1, a_2, \dots, a_n]$, 相邻整数由一个空格隔开, 表示你找到的一组合法的 a 序列;
- 若无解, 输出一行一个字符串 “No”。

“Yes” 和 “No” 均不区分大小写, 例如 “Yes”、“YES”、“yEs” 都被认为是肯定的结果, 而 “No”、“NO”、“n0” 都被认为是否定的结果。

样例

standard input	standard output
5	Yes
3	1 2 3
1 1 2 2 3 3	No
3	Yes
1 1 1 1 2 3	1 1 4 5 1 4
6	No
1 1 1 1 1 4 4 4 5 5 5 5	Yes
5	9 9 8 2 4 4 3 5 3
1 1 1 2 3 4 5 5 5 5	
9	
2 3 3 4 4 5 8 9 9 9 9 9 9 9 9 9 9	

题目 J. 翻转逆序对

tarjen 有一个长度为 n 的排列 p 。你可以执行至多一次操作：

- 选择两个整数 l 和 r ($1 \leq l \leq r \leq n$)，将子数组 $[p_l, p_{l+1}, \dots, p_r]$ 翻转。

聪明的 tarjen 想考考你，操作后排列最多有多少个逆序对？

术语说明：逆序对定义为满足 $1 \leq i < j \leq n$ 且 $p_i > p_j$ 的下标对 (i, j) 。

输入

第一行有一个整数 T ($1 \leq T \leq 10^5$)，表示数据组数。接下来 T 段，每段描述一组数据。对于每组数据：

- 第一行一个整数 n ($1 \leq n \leq 8000$) 表示排列长度。
- 第二行 n 个整数 $p_1, p_2, \dots, p_n$ ($1 \leq p_i \leq n$) 表示给出的排列。

数据保证 $\sum n \leq 8000$ 。

输出

对于每组数据，输出一行一个整数，表示最大逆序对数。

样例

standard input	standard output
3	2
3	10
2 1 3	10
5	
5 4 3 2 1	
6	
3 5 1 4 2 6	

注释

第一组数据中，翻转区间 $[1, 3]$ 得到 $[3, 1, 2]$ ，有 2 个逆序对： $(1, 2)$ 和 $(1, 3)$ 。

第二组数据中，不进行操作即可得到 10 个逆序对。

第三组数据中，翻转区间 $[3, 6]$ 得到 $[3, 5, 6, 2, 4, 1]$ ，有 10 个逆序对： $(1, 4)$ 、 $(1, 6)$ 、 $(2, 4)$ 、 $(2, 5)$ 、 $(2, 6)$ 、 $(3, 4)$ 、 $(3, 5)$ 、 $(3, 6)$ 、 $(4, 6)$ 和 $(5, 6)$ 。

题目 K. 拆分树

tarjen 在玩把两棵树拼成一个图的游戏。他用两棵 n 个点的树拼成了一个 n 个点、 $m = 2n - 2$ 条边的无向无权图 $G = (V, E)$ 。但是他发现他没法把这个图还原回两棵树了，请你帮帮他！

形式化地说，请你将所有边恰好分成两个集合 T 和 $E \setminus T$ ，使得 T 和 $E \setminus T$ 都构成 G 的一棵生成树。你可以输出任意一种合法方案。保证合法方案一定存在。

注意：图中可能存在重边，但保证不存在自环。

输入

第一行有一个整数 T ($1 \leq T \leq 5000$)，表示数据组数。接下来 T 段，每段描述一组数据：

- 第一行一个整数 n ($2 \leq n \leq 5000$)，表示点数。
- 接下来 $2n - 2$ 行，每行两个整数 u, v ($1 \leq u, v \leq n, u \neq v$)，表示一条边。边按输入顺序从 1 到 $2n - 2$ 编号。

数据保证 $\sum n$ 不超过 10000。

输出

对于每组数据，输出 $n - 1$ 个整数（升序），表示第一棵生成树包含的边的编号。剩余的 $n - 1$ 条边也应构成一棵生成树。

样例

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

注释

对于第一组样例，边 $\{1, 2\}$ 对应 $\{(1, 2), (2, 3)\}$ ，构成一棵生成树。剩余边 $\{3, 4\}$ 对应 $\{(1, 3), (1, 2)\}$ ，也构成一棵生成树。

对于第二组样例，边 $\{1, 2, 3\}$ 对应 $\{(1, 2), (2, 3), (3, 4)\}$ ，构成一棵生成树。剩余边 $\{4, 5, 6\}$ 对应 $\{(1, 4), (1, 3), (2, 4)\}$ ，也构成一棵生成树。

题目 L. 顺子

斗地主是中国最流行的扑克牌游戏之一。游戏使用一副 54 张的扑克牌（52 张普通牌 + 2 张王牌）。

在斗地主中，顺子是由至少 5 张连续点数的牌组成的牌型。点数从小到大依次为：3, 4, 5, 6, 7, 8, 9, 10, J, Q, K, A。注意，2 和大小王不能出现在顺子中。

现在 *tarjen* 手上有 17 张牌，他很喜欢顺子。笨蛋 *tarjen* 想知道他能打出的最长顺子的长度，但是他不会数数，能请你帮帮他吗？

如果无法打出任何顺子，请输出 0。

输入

第一行有一个整数 T ($1 \leq T \leq 1000$)，表示数据组数。接下来 T 行，每行描述一组数据。

每组数据包含 17 个由空格分隔的字符串，表示 17 张牌。每张牌用以下方式表示：

- 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K, A：分别表示对应点数的牌；
- 2：表示 2；
- SJ：表示小王；
- BJ：表示大王。

保证每种普通牌（A, 2~10, J, Q, K）每组数据中最多出现 4 次，小王和大王各最多出现 1 次。

输出

对于每组数据，输出一行一个整数，表示能打出的最长顺子的长度。如果无法打出顺子（即不存在长度 ≥ 5 的连续点数序列），输出 0。

样例

standard input	standard output
2	5
3 4 5 6 7 3 4 5 6 7 2 2 2 2 SJ BJ A	0
2 2 2 2 3 3 3 3 4 4 4 4 5 5 5 5 7	

注释

第一个样例中，手中可用于顺子的点数有 3, 4, 5, 6, 7 和 A，其中最连续序列为 3, 4, 5, 6, 7，长度为 5。

第二个样例中不存在合法的顺子，答案为 0。

题目 M. 水晶城堡

$tarjen$ 是水晶城堡的守护者。城堡的长廊里镶嵌着一排 n 颗魔法水晶，第 i 颗水晶的颜色编号为 a_i 。长廊中相邻且同色的水晶会产生共鸣，形成一个**色段**——即极大的连续相同颜色段。例如颜色序列 $[1, 1, 2, 2, 1]$ 有 3 个色段： $[1, 1]$ 、 $[2, 2]$ 和 $[1]$ 。

每天都有旅行者慕名前来，提出 q 个问题。每个问题指定一段区间 $[l, r]$ ：如果把这段水晶取下来随机打乱重新排列（所有不同的颜色序列等概率出现），形成的色段数量的期望值是多少？

答案对 998244353 取模。即若答案为最简分数 $\frac{x}{y}$ ，输出 $x \cdot y^{-1} \bmod 998244353$ 。可以证明在本题约束下 y^{-1} 总是存在的。

输入

第一行有一个整数 T ($1 \leq T \leq 10^5$)，表示数据组数。接下来 T 段，每段描述一组数据：

- 第一行两个整数 n, q ($1 \leq n, q \leq 10^5$) 表示水晶数量和询问次数。
- 第二行 n 个整数 $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq n$) 表示每颗水晶的颜色编号。
- 接下来 q 行，每行两个整数 l, r ($1 \leq l \leq r \leq n$) 表示询问区间的端点。

数据保证 $\sum n \leq 10^5$ ， $\sum q \leq 10^5$ 。

输出

对于每组数据中的每个询问，输出一行一个整数，表示期望色段数量对 998244353 取模的结果。

样例

standard input	standard output
1 4 2 1 1 2 2 1 2 1 4	1 3
1 10 5 3 5 3 3 6 4 8 2 3 5 6 9 1 8 8 10 4 9 7 7	4 748683272 3 665496241 1

注释

对于第一组样例：

第一个询问，取出的水晶颜色为 $[1, 1]$ ，只有一种排列，色段数为 1。

第二个询问，取出的水晶颜色为 $[1, 1, 2, 2]$ 。6 种排列的色段数分别为 2, 4, 3, 3, 4, 2，期望值为 $\frac{18}{6} = 3$ 。