

Problem A. 炼金术士

为了炼制贤者之石，耶芙娜正在精心布置她的炼金阵，炼金阵由 n 个点的无向完全图组成，即任意两个不同的点之间连有一条无向边。每一条边可以选择进行黑化，或者白化，或者不进行任何处理。

炼金阵对于黑化边与白化边构成的子图有一定的要求：

- 在黑化边和白化边各自构成的子图中，都不能出现三元环。即不存在三个点之间两两连边全部黑化或全部白化；
- 黑化边构成的子图中需要存在一条边数为 k 的无环链作为基底，这条链经过的 $k + 1$ 个点 a_0, a_1, \dots, a_k 将会在输入中给出，对于任意 $i \in [0, k - 1]$ ，炼金阵中 a_i 与 a_{i+1} 两个点的连边必须进行黑化。

在满足以上两个要求的同时，为了提高炼制的成功率，耶芙娜希望黑化边与白化边的数量之和尽可能大。作为耶芙娜的老板（兼宠物猫），你决定帮助她规划出一种黑化白化边数量之和最大的炼金阵图。

Input

第一行一个整数 T ($1 \leq T \leq 100$)，表示测试数据组数。

对于每组数据，第一行输入两个整数 n, k ($2 \leq n \leq 50, 1 \leq k \leq n - 1$)，表示炼金阵点数与基底链的长度。第二行输入 $k + 1$ 个互不相同的整数 a_0, a_1, \dots, a_k ($1 \leq a_i \leq n$)，表示黑化基底链经过的点的编号。

Output

对于每组数据，你需要输出 n 行。第一行输出黑化边与白化边数量之和的最大值。接下来 $n - 1$ 行每行输出一个由 +, -, 0 三种字符构成的字符串，分别表示黑化、白化和不做处理，第 i 行长度为 i ，第 i 行第 j 个字符表示第 $i + 1$ 个点与第 j 个点连边的处理类型。

答案可能不唯一，你可以输出任意最优解，但需要保证输出中基底链上的边均为 +，且 + 与 - 字符数量之和等于第一行输出的最大值。

Example

standard input	standard output
3	6
4 2	+
1 2 3	--+
5 4	--+
1 3 5 2 4	10
6 5	-
1 2 3 4 5 6	+ +- ++- -++- 14 + -+ --+ +--+ 0+--+

Problem B. 简单可满足性问题

布尔可满足性问题 (SAT) 问题的定义如下:

有 n 个变量 x_1, \dots, x_n , 每个变量可以取值 0 (假) 或 1 (真), 每个变量有正文字 x_i 与负文字 \bar{x}_i 两种形式。其中正文字 x_i 的值与变量相同, 负文字 \bar{x}_i 的值与变量相反。

一个子句为若干个文字相析取, 即 $C_i = x_{a_{i,1}} \vee \dots \vee x_{a_{i,p}} \vee \bar{x}_{b_{i,1}} \vee \dots \vee \bar{x}_{b_{i,q}}, 1 \leq a_{i,j}, b_{i,k} \leq n$ 。这些文字中有任意一个文字的取值为 1, 则该子句的值为 1; 反之若所有文字的取值都为 0, 则该子句的值为 0。

一个合取范式 (CNF) 为 m 个子句 C_1, \dots, C_m 相合取, 即 $C = C_1 \wedge \dots \wedge C_m$ 。若所有子句的取值为 1, 则该 CNF 的值为 1; 反之若任意一个子句的取值为 0, 则该 CNF 的值为 0。

SAT 问题即给出一个 CNF, 问存不存在一种对变量的赋值, 使该 CNF 的值为 1。

Bob Wang 知道要找到一组让所有子句都为真的变量赋值是一件很困难的事情, 因此对于给出的 CNF $C_1 \wedge \dots \wedge C_m$, Bob Wang 只需要找出一种变量赋值, 使得至少 $\lceil \frac{m}{2} \rceil$ 个子句的赋值为 1。可以证明满足要求的赋值一定存在。

Input

第一行两个整数 n, m ($1 \leq n, m \leq 10^6$), 表示变量个数以及子句个数。

接下来 m 组数据, 描述子句。

对于第 i 个子句, 第一行一个整数 c_i ($1 \leq c_i \leq 10^6$), 表示文字的数量。接下来 c_i 行, 每行两个整数 x, y , 描述该子句包含的文字。其中 x 表示变量编号, $y = 1$ 表示正文字 x , $y = 0$ 表示负文字 \bar{x} 。

保证 $\sum_{i=1}^m c_i \leq 10^6$ 。

Output

一行 n 个整数, 表示变量的赋值。其中 $x_i = 1$ 表示第 i 个变量赋值为 1, $x_i = 0$ 表示第 i 个变量赋值为 0。输出使至少 $\lceil \frac{m}{2} \rceil$ 个子句的赋值为 1 的任意一组赋值即可。

Example

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

Note

样例表示的子句为 $C_1 = x_1, C_2 = x_2, C_3 = \bar{x}_3, C_4 = x_4, C_5 = x_1 \vee x_3$, 容易证明 $x_1 = 0, x_2 = 1, x_3 = 0, x_4 = 1$ 是符合题意的一组赋值。

Problem C. 箭串

定义「箭串」为满足以下条件的串:

- 串的长度至少为 5;
- 串以 > 开始, 并以 >>> 结尾;
- 串的其余部分仅由 - 组成。

例如, >-->>> 和 >--->>> 是合法的箭串, 而 >->> 和 >->->>> 不是。

现有一个长度为 n 且最初仅包含字符 * 的字符串, Sauden 会对这个串进行 m 次操作, 每次操作中, 他会选择这个串中一个长度大于等于 5 的子串, 并用一个等长的箭串替代这个子串。请你输出 m 次操作后这个串会变成什么。

Input

第一行一个整数 T ($1 \leq T \leq 10^4$), 表示测试数据组数。

对于每组数据, 第一行两个整数 n 和 m ($5 \leq n \leq 10^5, 0 \leq m \leq 10^5$), 分别表示初始串长度和操作数。

接下来 m 行, 第 i 行两个整数 p 和 l ($1 \leq p \leq n - 4, 5 \leq l \leq n, p + l - 1 \leq n$), 表示第 i 次操作中选择的子串是从第 p 个字符开始, 并且长度为 l 。即选择的子串为 $[p, p + l - 1]$ 。

对于所有数据, 满足 $\sum n \leq 5 \cdot 10^5, \sum m \leq 5 \cdot 10^5$ 。

Output

对于每组数据输出一行一个字符串, 表示所有操作之后的字符串。

Example

standard input	standard output
3	*****
6 0	>->>>*
6 1	>>->>>
1 5	
6 2	
1 6	
2 5	

Problem D. 阿罗祖斯坦的桥

Natsuzora 的故乡，阿罗祖斯坦（Arozustan），是一个二维空间中的国家。

在阿罗祖斯坦，共有 $n + 1$ 座城市，分布在一条直线上。为了进行新一代的交通，Natsuzora 计划在阿罗祖斯坦修建大量桥梁。

由于次元不同导致的一些我们无法理解的原因，阿罗祖斯坦的桥梁必须以固定的曲率 k 修建，并且必须位于直线上方。更直白地说，这些桥梁形如一段连接两个城市的圆弧，且圆弧所对应的整圆的直径是固定的。除此之外，桥梁之间不能够有交叉的地方，即桥梁与桥梁之间只能够在每个城市处有公共点。

为了最大化阿罗祖斯坦的交通效率，Natsuzora 希望修建的桥梁越多越好。在此基础上，为了尽可能地节省经费，还需要使得桥梁的总长度最小。由于二维空间中难以组织足够的算力，Natsuzora 向你求助，希望你能帮他求出桥梁总长度的最小值。

Input

第一行包含两个整数 n, d ($1 \leq n \leq 2 \times 10^3, 1 \leq d \leq 10^6$)，分别表示城市之间间隔的数量和曲率 k 所对应的圆的直径。

第二行包含 n 个整数 a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^4$)，其中 a_i 表示从左到右第 i 个城市与第 $i + 1$ 个城市之间的距离。

保证 $\sum a_i \leq d$ 。

Output

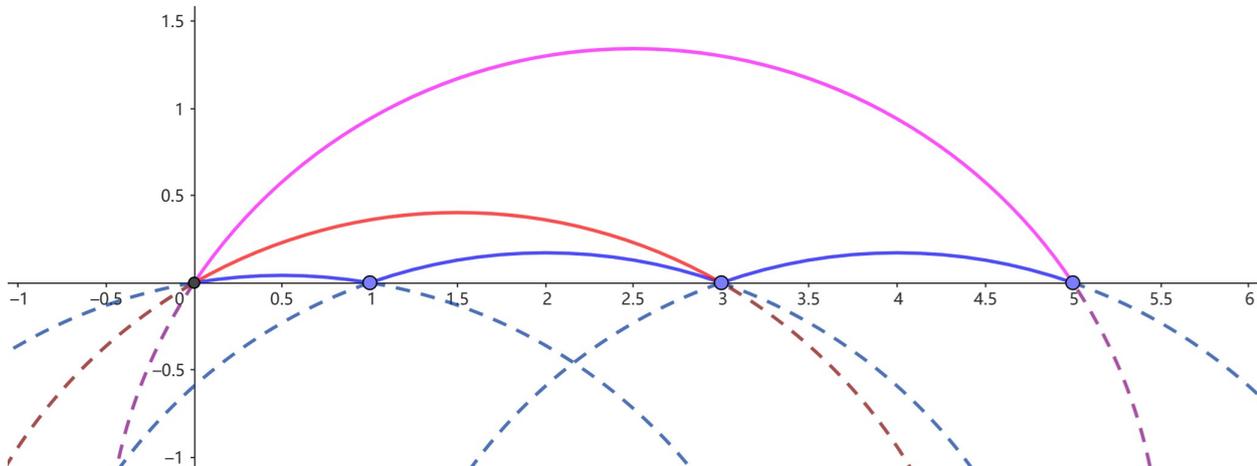
输出一个数字，表示桥梁总长度的最小值。当与标准答案之间的相对误差不超过 10^{-9} 时，你的答案被视为正确。

Example

standard input	standard output
3 6 1 2 2	14.134988742384

Note

样例图示如下：



Problem E. 猫猫城大选 (困难版)

本题的简单版是 2024 年趣味赛第一场 C 题, 该版本与简单版的区别在于猫猫虫的类型、数据范围以及输出要求的不同。

在遥远的大洋彼岸, 矗立着一座猫猫城, 猫猫城里住着许多猫猫虫。

和人类社会一样, 猫猫虫也是有领导猫的, 并且每隔一段时间都会举行一场盛大的猫猫城大选。今年的大选即将开始。

猫猫城的现任领导猫, Capoo, 希望能够蝉联猫猫城领导猫的席位, 因此它想在大选开始前进行一次民意调查。

在此次民意调查中, 猫猫城的每个公民都拥有投票的权利, 它们可以根据自己的意愿选择是否支持 Capoo 继续担任猫猫城的领导猫。Capoo 希望自己的支持率大于 50%。

有的猫猫虫有自己坚定的想法, 然而让有的猫猫虫做出自己的决定是一件很困难的事情, 因此它们都会有从众心理或者逆反心理。具体地, 现在猫猫城里一共有 n 只猫猫虫 (不包含 Capoo, 很显然 Capoo 并不能给自己投票), 对于第 i 只猫猫虫, 它将属于以下四种类型当中的一种:

1. 无条件投出支持票;
2. 无条件投出反对票;
3. 如果支持数大于反对数, 那么它将投下支持票; 如果支持数小于反对数, 那么它将投下反对票; 如果支持数等于反对数, 那么它将有 p_i 的概率投下支持票, $1 - p_i$ 的概率投下反对票;
4. 如果支持数大于反对数, 那么它将投下反对票; 如果支持数小于反对数, 那么它将投下支持票; 如果支持数等于反对数, 那么它将有 p_i 的概率投下支持票, $1 - p_i$ 的概率投下反对票。

Capoo 希望能够预测调查的结果, 因此它前来求助大魔法师波波王。波波王使用读心术看出了每一只猫猫虫属于哪种类型。现在 Capoo 想知道, 在所有 n 只猫猫虫按顺序投完票后, 支持数减去反对数的期望是多少, 对 998244353 取模。

Input

第一行一个整数 n ($1 \leq n \leq 10^6$), 表示猫猫城里猫猫虫的数量, 不包含 Capoo 自己。

接下来 n 行, 第 i 行两个数, 描述第 i 只投票的猫猫虫。一个整数 a_i ($a_i \in \{1, 2, 3, 4\}$), 表示第 i 只猫猫虫属于上述描述中的哪个类型; 一个整数 p_i ($0 \leq p_i < 998244353$), 题目中保证所有出现的概率为有理数, 记与这只猫猫虫有关的概率为 $\frac{x_i}{y_i}$, 其中 x_i 为非负整数, y_i 为正整数, 则给出的 p_i 为 $x_i \cdot y_i^{-1} \bmod 998244353$, 其中 y_i^{-1} 表示 y_i 在模 998244353 意义下的逆元。保证若 $a_i = 1$, 则 $p_i = 1$; 若为 $a_i = 2$, 则 $p_i = 0$ 。

Output

一行一个整数, 表示支持数减去反对数的期望, 对 998244353 取模。可以证明这个期望是一个有理数, 记为 $\frac{p}{q}$, 你要输出的答案为 $p \cdot q^{-1} \bmod 998244353$, 其中 q^{-1} 表示 q 在模 998244353 意义下的逆元。

Example

standard input	standard output
2	0
3 1	
4 114514	

Note

对于样例, 显然第一只猫猫虫一定会投下支持票, 第二只猫猫虫一定会投下反对票, 因此支持票减去反对票的数量为 0。

Problem F. 我知道你姓啥!

给定 n 个长度为 m 的二进制字符串 s_1, s_2, \dots, s_n , 每个字符串由字符 0 和 1 组成。存在一个神秘数字 x ($1 \leq x \leq m$), 对于每个字符串 s_i , 你可以得知 $s_i[x]$ 的值。

你的任务是确定至少需要添加多少个长度为 m 的二进制字符串到现有集合中, 使得通过查询所有字符串 (包括添加的字符串) 的 $s_i[x]$ 值, 能够唯一确定神秘数字 x 。

Input

第一行一个整数 T ($1 \leq T \leq 10^4$), 表示测试数据的组数。

对于每组数据, 第一行两个整数 n 和 m ($1 \leq n, m \leq 10^6, n \cdot m \leq 10^6$), 分别表示二进制字符串的数量和每个字符串的长度。

接下来 n 行, 每行包含一个长度为 m 的二进制字符串 s_i 。

保证所有数据的 $n \cdot m$ 之和不超过 10^6 。

Output

对于每组数据, 输出一个整数, 表示至少需要添加的二进制字符串的数量, 以确保能够唯一确定神秘数字 x 。

Example

standard input	standard output
2	1
2 6	0
111000	
101010	
2 4	
1100	
1010	

Note

第一组样例中, 我们可以添加字符串 $s_3 = 001001$, 当我们询问得知 $s_1[x] = 0, s_2[x] = 0, s_3[x] = 1$ 的时候我们可以确定 $x = 6$ 。当 x 为其他值的时候也可以类似猜中。

Problem G. 猜数游戏

你正在玩一个猜数游戏。每轮游戏给定 a, p, l, r ，你需要猜一个 $[l, r]$ 区间内的正整数 x ，满足 $x \bmod p = a$ 。但是由于程序 bug，你不知道 a, p 的值而只知道 l, r 的值。

你不会 Pwn，所以这个游戏对你来说太难了。但是好在有一天有人托梦给你，你知道了存储 a 和 p 值的地址，进而你知道了每轮中这两个数的值。请根据现有信息进行这个猜数游戏。

Input

第一行一个整数 T ($1 \leq T \leq 10^4$)，表示游戏轮数。

对于每轮游戏，一行四个整数 a, p, l, r ($0 \leq a < p \leq 10^{18}, 1 \leq l \leq r \leq 10^{18}$)，表示现有信息。

Output

对于每轮游戏，如果不存在这样的 x ，输出一行 No，否则先输出一个字符串 Yes，再在同一行空一格后输出一个正整数 x ($l \leq x \leq r$)，表示你猜的数。

Example

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

Problem H. 独立事件

设有一个不均匀的 n 面骰子, 其第 i 面朝上的概率为 $\frac{a_i}{1000}$ 。定义事件 A 为「骰子朝上的面属于集合 $\{1, 2, \dots, k\}$ ($k = 0$ 时集合为空集)」。对于任意子集 $I \subseteq \{1, 2, \dots, n\}$, 定义事件 B_I 为「骰子朝上的面属于集合 I 」。试求在全部 2^n 个事件 B_I 中, 满足以下独立性条件

$$P(A) \cdot P(B_I) = P(A \cap B_I)$$

的事件数量。

Input

第一行两个整数 n 和 k ($1 \leq n \leq 1000, 0 \leq k \leq n$)。

第二行 n 个整数 a_1, a_2, \dots, a_n ($1 \leq a_i \leq 1000$), 保证 $\sum_{i=1}^n a_i = 1000$ 。

Output

输出一行一个整数, 表示答案。由于这个值可能很大, 你需要输出答案对 998244353 取模后的值。

Examples

standard input	standard output
4 2 200 300 200 300	4
1 1 1000	2
3 0 1 997 2	8

Note

第一组样例中满足条件的 I 有 $\emptyset, \{1, 3\}, \{2, 4\}, \{1, 2, 3, 4\}$ 四个。

第三组样例 $k = 0$, 此时对于任何事件 B_I 均满足 $P(A) \cdot P(B_I) = P(A \cap B_I) = 0$, 满足条件的事件 B_I 共有 $2^3 = 8$ 个。

Problem I. 圆与直线交点

给定二维平面上不共线的三点 a, b, c 。这三点可以确定一个圆 O 。

给定直线上一点 p ，以及直线的方向向量 v 。询问该直线与圆 O 的位置关系。

Input

第一行一个整数 t ($1 \leq t \leq 10^4$)，表示数据组数。

对于每组数据，共五行。每行两个整数 x_i, y_i ($-10^9 \leq x_i, y_i \leq 10^9$)，分别表示 a, b, c, p, v 的坐标。保证 a, b, c 三点不共线，并且直线的方向向量不是零向量。

Output

如果相交输出 Yes，相离输出 No。特别地，当直线与圆相切的时候输出 Or。

Example

standard input	standard output
3	Or
0 0	Yes
0 1	No
1 0	
1 0	
1 1	
0 0	
0 1	
1 0	
1 1	
1 1	
0 0	
0 1	
1 0	
2 0	
0 1	

Problem J. 创建用户

现有 n 个人要入职，你需要按这个顺序为这些人注册办公账号，因此你需要为这些人安排用户名。设定用户名的规则是保留姓名中第一个字的全拼，取剩余字的拼音首字母附在后面。如果当前用户名与之前的某个用户名相同，则在后面加上 1, 2..... 依次类推。

Input

第一行一个整数 n ($1 \leq n \leq 1000$)。

接下来 n 行，每行若干个字符串，字符串之间用一个空格隔开，表示此人的姓名。保证每行字符串个数在 $[2, 4]$ 范围内，每个字符串长度在 $[2, 6]$ 范围内，且仅包含小写英文字母。

Output

输出 n 行，对于每个人输出对应的用户名。

Example

standard input	standard output
6	liub
liu bei	liub1
liu biao	lius
liu shan	liuss
liu shan shan	lius1
liu shen	zhugl
zhu ge liang	

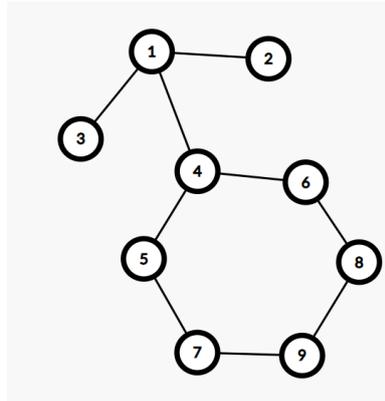
Problem K. 哲学之门

在兄贵们居住的异次元幻想乡——新日暮里，比利王和香蕉君两位哲学家正面临一场危机！

暗黑大将军 VAN 样通过奥义「Deep ∂ Dark ∂ Fantasy」污染了哲学能量核心，导致整个世界的男魂符号「∂」陷入混乱。为了拯救新日暮里，两位哲学家想要去穿越那扇藏在「VANの神殿」深处的「哲学之门」，传说只有集齐所有「哲学密钥」的兄贵，才能解开「哲学之门」的封印，获得「东方馅挂炒饭」的终极配方，阻止暗黑大将军的阴谋。然而，这扇门的钥匙被分割成碎片，散落在「VANの神殿」的迷宫中！

危难之际，两位哲学家打破了第 ⑨ 面墙，在电子科技大学这块充满灵性的土地上，找到了热衷于 xpc 竞赛的你，相信拥有高超计数能力的你，一定可以在迷宫中找齐所有「哲学密钥」，用「东方馅挂炒饭」拯救陷入混乱的幻想乡！

「VANの神殿」的迷宫是一张由 n 个点 m 条边组成的无向图，这张图中的每个点度数不超过三（即与每个点相邻的边不超过三条），且保证不出现重边与自环。你需要统计出这张图中形如「哲学密钥」的连通子图数量。「哲学密钥」是一个包含九个点九条边的连通子图，且九个点需要按照如图所示的位置关系进行连边（图中点的编号仅做参考）。



注意：每条边和每个点可以属于多个不同的「哲学密钥」子图。两个连通子图被视为不同的「哲学密钥」当且仅当两个子图中的边集不完全相同。

Input

第一行一个整数 T ($1 \leq T \leq 10^4$)，表示测试数据组数。

对于每组数据，第一行输入两个整数 n, m ($2 \leq n \leq 10^5, 1 \leq m \leq 1.5n$)，表示迷宫的点数与边数，接下来 m 行每行输入两个整数 x, y ($1 \leq x, y \leq n$)，表示 x 与 y 之间有一条无向边，数据保证 $x \neq y$ 。

保证所有测试数据 n 之和不超过 10^5 。

Output

对于每组数据，你需要输出一行一个整数，表示不同的「哲学密钥」连通子图数量。

Example

standard input	standard output
2	1
11 12	2
1 2	
2 3	
3 4	
4 5	
5 6	
6 1	
6 7	
7 8	
7 9	
10 11	
10 3	
10 2	
9 11	
1 2	
2 3	
3 4	
4 5	
5 6	
6 1	
6 7	
7 8	
7 9	
5 9	
8 9	

Problem L. 乒乓

有 n 个人前来打乒乓。

由于时间有限, 一共只能打 m 场, 你需要告诉他们一共有多少种安排方式, 使得恰好有 k 个人只打了一场比赛, 答案对 998244353 取模。

具体的, 一种安排由 m 场比赛组成, 每场比赛由两个不同的人完成。且在一个安排中, 两个人之间不能打多次。可以有人一场也不打。两种安排不同, 当且仅当存在两个人, 他们在一个安排中打了, 而在另一个安排中没有。

Input

第一行一个正整数 T ($1 \leq T \leq 1000$), 表示数据组数。

对于每组数据, 一行三个整数, 分别为 n, m, k ($1 \leq n \leq 5 \times 10^3, 0 \leq m \leq \frac{n \times (n-1)}{2}, 0 \leq k \leq n$)。

保证对于所有数据, 满足 $\sum n \leq 5 \times 10^3$ 。

Output

对于每组数据, 输出一行一个整数, 表示答案。

Example

standard input	standard output
2	12
4 3 2	3
3 2 2	

Problem M. 前兜车一转后兜车一转

给定一个长为 n 的正整数序列 a , 序列的价值定义为 $\sum_{i=1}^{n-1} |a_{i+1} - a_i|$ 。

现在你需要对序列的一个非空前缀进行翻转, 再对序列的一个非空后缀进行翻转, 使得序列的价值最大。

称 p 为 a 的一个前缀, 当且仅当 p 可以通过 a 从后往前依次删除若干元素得到, 称 s 为 a 的一个后缀, 当且仅当 s 可以通过 a 从前往后依次删除若干元素得到。例如, 对于 $a = \{1, 2, 3, 4, 5\}$, 则 $\{1, 2, 3\}, \{1\}, \{1, 2, 3, 4, 5\}$ 均为 a 的前缀, 而 $\{2, 3, 4\}, \{1, 3, 4\}$ 不是 a 的前缀; $\{4, 5\}, \{5\}, \{1, 2, 3, 4, 5\}$ 均为 a 的后缀, 而 $\{2, 3, 5\}, \{3, 4\}$ 不是 a 的后缀。

翻转操作按如下方式进行:

1. 对于要翻转的区间 $[l, r]$, 定义 $b_i = a_{r-i+1}$, 其中 $1 \leq i \leq r - l + 1$;
2. 用得到的 b 序列替换 a 序列中 $[l, r]$ 位置的元素。

例如, 对于 $a = \{1, 2, 3, 4, 5\}$, 翻转区间为 $[1, 3]$, 得到的翻转后数组为 $a = \{3, 2, 1, 4, 5\}$ 。

Input

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

对于每组数据, 第一行一个正整数 n ($1 \leq n \leq 10^5$), 表示序列的长度。

第二行 n 个正整数 a_i ($1 \leq a_i \leq 10^9$), 表示给定的序列。

对于所有数据, 满足 $\sum n \leq 10^6$ 。

Output

对于每组数据, 输出一行一个整数表示序列两次翻转后最大的价值。

Example

standard input	standard output
5	0
1	4
1	13
2	19
3 7	21
3	
10 2 7	
5	
1 5 3 8 7	
8	
9 7 3 6 8 5 4 6	

Problem N. 幸运之环 II

Kanade 收到了一些由 n 个节点和 n 条边组成的无向连通图，并且图中无重边和自环。容易发现这个图中有且只有一个简单环。若存在一个点序列 v_0, v_1, \dots, v_k ，满足 v_i 与 v_{i+1} ($0 \leq i < k$) 之间有一条边， $v_0 = v_k$ 且 v_1, \dots, v_k 互不相同，则称图中存在一个环，环上的节点为 v_1, \dots, v_k 。

Kanade 认为这些图中的环是「幸运」的，因此她会从图中取出送给 Isla 当礼物。请帮 Kanade 找出每个图送给 Isla 的环。

Input

第一行一个正整数 T ($1 \leq T \leq 300$)，表示 Kanade 收到的图的个数。

对于每张图，第一行一个正整数 n ($3 \leq n \leq 10^5$)，表示图的点数。

接下来 n 行每行两个整数 u, v ($1 \leq u, v \leq n, u \neq v$)，表示有一条连接 u, v 两个点的无向边。保证给出的图无重边。

保证所有图的 n 之和不超过 5×10^5 。

Output

对于每张图输出一行，首先输出一个整数 k ，代表环上的节点个数，接下来输出 k 个整数，按顺时针或逆时针方向输出环上的节点编号。**你需要输出字典序最小的一组答案。**

对于两个长度为 l 的序列 a 和 b ，称 a 的字典序比 b 小，当且仅当存在 p ($1 \leq p \leq l$)，满足 $a_i = b_i$ ($1 \leq i < p$)，且 $a_p < b_p$ 。

Example

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

Problem O. 不走回头路

去年的某个时候, Bob Wang 来到云南大理旅游。平静的洱海水面上, 一只只海鸥从低空飞过。成群的野鸭划过水面, 在这镜面上留下了一排整齐的划痕。微风拂起, 水面上泛起一阵阵涟漪。洱海西面则是大理古城。古城的夜晚人声鼎沸, 灯火通明。街上, 各种小商贩在叫卖, 有糯糯的烤饵块、酸辣清爽的拌水果、奶香十足的烤乳扇……游走在盛景与古刹之间, 在大理, 每个人都可以找到自己的归宿, 感受到生命的平静与喜悦。

但是在大街小巷游走的时候, Bob Wang 了解到当地的一个很有趣的习俗。当地人在游览各处时, 会避免走回头路。Bob Wang 根据自己的理解, 希望制定一条游览路线。具体地, 一共有 n 个景点, 这些景点之间有 m 条道路连接。Bob Wang 给每条道路指定了一个方向, 在走到这条路时只能沿着指定的方向走。Bob Wang 希望在这些限制上, 找到一个游览方案 $s = (s_1, \dots, s_k)$, 满足 $(s_i, s_{i+1}), 1 \leq i < k$ 为 Bob Wang 指定的某条有向道路, 同时 $\{1, \dots, n\} \subseteq \{s_1, \dots, s_k\}$ 。即 Bob Wang 可以从景点 s_1 开始, 游览完所有景点。请告诉 Bob Wang 是否存在这样的游览方案。

Input

第一行两个整数 n, m ($2 \leq n \leq 10^6, n-1 \leq m \leq 10^6$), 表示景点的数量以及道路的数量。保证景点之间相互连通。

接下来 m 行, 第 i 行两个整数 u_i, v_i ($1 \leq u_i, v_i \leq n, u_i \neq v_i$), 表示景点 u_i 和 v_i 之间存在一条道路, 并且 Bob Wang 指定该条道路只能从 u_i 走到 v_i 。一对景点之间最多有一条道路。

Output

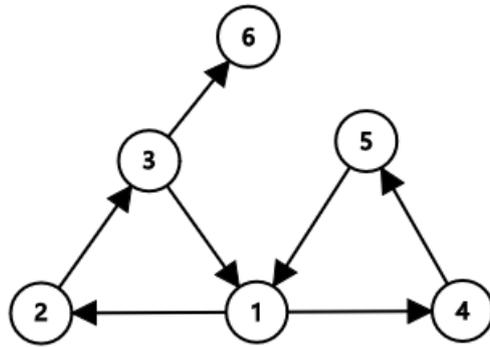
一行 Yes 或 No, 表示是否存在符合题意的方案。

Examples

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

Note

样例 1 如图所示, 一种可行的游览方案为 $(2, 3, 1, 4, 5, 1, 2, 3, 6)$ 。



Problem P. 相同字符

在文本预处理和模式识别中，快速验证字符串变换可行性是常见需求。本题要求处理多组测试数据，验证能否通过删除原字符串中的部分小写字母（保留可能存在的关键大写字母），使处理后的字符串与目标字符串在忽略大小写的情况下完全一致。

给定字符串对 (a, b) ，其中：

- 字符串 b 仅由小写字母构成；
- 字符串 a 由小写字母和至多一个大写字母构成。

对于每组数据，判断是否可以通过删除 a 中的若干个小写字母（允许删除零个，但不可删除大写字母），使得处理后的字符串 a' 与 b 满足：

$$\text{lower}(a') = \text{lower}(b)$$

其中 $\text{lower}(\cdot)$ 表示将字符串转换为全小写形式。

Input

第一行一个整数 T ($1 \leq T \leq 10^3$)，表示测试数据组数。

对于每组数据，共两行。第一行一个字符串 a ($1 \leq |a| \leq 2 \times 10^5$)，第二行一个字符串 b ($1 \leq |b| \leq 2 \times 10^5$)。保证字符串 a 中最多出现一个大写字母，其余均为小写字母，保证字符串 b 中仅出现小写字母。

对于所有测试数据，保证 a 和 b 字符串总长度之和不大于 10^6 。

Output

对于每组数据，若存在可行方案，输出 **Yes**；否则输出 **No**。

Example

standard input	standard output
6	Yes
Aa	Yes
a	No
aaaB	No
ab	Yes
aAb	Yes
ba	
uEstc	
ustc	
Uestc	
ustc	
abcde	
ace	

Problem Q. 校车

从 CTSEU 毕业之后, wwlw 去 CTSEU 附属小学当了校长。为了保证同学们上学准时且安全, 他决定开设校车接送同学们上学。

学校附近的城市路网可以用一个 n 个路口 m 条双向道路的连通图表示。每条双向道路连接了两个路口, 且有一个时间权重表示综合各种因素之后一辆校车通过该路段所需要的时间。在经过考量之后, wwlw 选择了其中 w 个路口作为校车的上车地点, 同学们按时在这些地点等待校车。

但是由于资金问题, 学校至多开设 K 条校车线路。每条校车线路都是一条从学校出发, 最后回到学校的一条闭合路径。每天早上, 所有校车会同时从学校出发, 沿着各自的路径运行, 接送沿途的同学, 最后回到学校。

现在, wwlw 需要你来规划这些线路, 使得这些线路覆盖了所有的上车地点, 且最晚回来的校车用时要尽可能少。你只需要告诉他最少用时。

Input

第一行四个整数 n, m, w, K ($2 \leq n \leq 500, 1 \leq m \leq \frac{n(n-1)}{2}, 1 \leq w \leq \min(n-1, 14), 1 \leq K \leq w$)。

接下来 m 行, 每行三个整数 u, v, c ($1 \leq u, v \leq n, 1 \leq c \leq 10^5, u \neq v$), 表示一条双向道路。保证图中无重边, 且给出的图是连通图。

接下来一行 w 个整数 p_1, p_2, \dots, p_w ($2 \leq p_i \leq n$), 表示选择的路口, 保证 p_1, \dots, p_w 两两不同。

约定一号路口是学校, 且保证上车地点不会选定在此处。

Output

输出一行一个整数, 表示最少用时。

Examples

standard input	standard output
5 5 4 1 1 2 1 2 3 2 3 4 3 4 5 4 1 5 5 2 3 4 5	15
5 5 3 2 1 2 2 2 4 2 1 3 100 3 4 1 4 5 3 2 3 5	14