

Problem A. Spacy 语言 (spacy)

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

这个题目需要你將一份 Spacy 语言的代码进行格式化。以下所有集合都指字符串集合。定义 $A \oplus B$ 为两个字符串 A 和 B 拼接形成的新字符串。

先给出一些定义:

- ϵ 表示空字符串。
- $c(A, B)$ 表示集合 $\{S \oplus T | S \in A, T \in B\}$ 。
- $C(A_1, A_2, \dots, A_n)$ 表示集合 $c(A_1, c(A_2, c(\dots, A_n)))$ 。
- $R(A, l, r)$ 表示 $\bigcup_{l \leq k \leq r} A^k$, 其中, $A^0 = \{\epsilon\}, A^k = \{S \oplus T | S \in A^{k-1}, T \in A\}$ 。
- $r(A, l)$ 表示 $\bigcup_{l \leq k} A^k$ 。
- 下文中, "以下之一"表示接下来所给出的集合的并集。
- $A + B$ 代表 $c(A, B)$ 。 $A + B + D$ 代表 $C(A, B, D)$, 以此类推。

S.P.A.C.Y. 发现了一个古老的编程语言, 叫做 Spacy 语言。设 W 为所有仅由空格 (ASCII 码为 32)、制表符 (ASCII 码为 9) 以及换行符 (ASCII 码为 10) 构成的非空字符串构成的集合; $S = L = E = W \cup \{\epsilon\}$ 。对于最终问题, 这些集合会与上面的定义有些区别。

设 T 为拉丁字母和下划线构成的集合。设 D 为 10 个数字字符构成的集合。设 U 为所有可见字符 (除去 "/") 以及空格、制表符、换行符构成的集合。

标识符是 $c(T, r(T \cup D, 0)) \setminus \{\text{if, otherwise, repeat, while, return}\}$ 。

数字是 $r(D, 1)$ 。

注释是 $C(/*, r(U, 0), */)$ 。

声明是 标识符 + W + 标识符。

表达式是以下之一:

1. 标识符;
2. 数字;
3. 表达式 + S + $\{+, \&, /, ==\}$ + S + 表达式;
4. 表达式 + S + $\{=\}$ + S + 表达式;
5. 标识符 + E + $\{(\} + E + \{)\}$;
6. 标识符 + E + $\{(\} + E +$ 表达式 + $E + \{)\}$;
7. 标识符 + E + $\{[\} + E +$ 表达式 + $E + \{]\}$;
8. 表达式 + E + $\{, \}$ + S + 表达式;
9. $\{*\}$ + E + 表达式;
10. $\{(\} + E +$ 表达式 + $E + \{)\}$;
11. 标识符 + E + $\{++\}$;

12. 表达式 $+ E + \{!\}$ 。

块 是 $\{\{\} + L + r(\text{陈述} + L, 0) + \{\}\}$ 。

条件 是 $\{(\} + E + \text{表达式} + E + \{)\}$ 。

陈述 是以下之一：

1. 声明 $+ E + \{?\}$;
2. 声明 $+ S + \{=\} + S + \text{表达式} + E + \{?\}$;
3. 表达式 $+ E + \{?\}$;
4. $\{\text{if}\} + S + \text{条件} + S + \text{块} + r(S + \{\text{otherwise}\} + W + \{\text{if}\} + S + \text{条件} + S + \text{块}, 0) + R(S + \{\text{otherwise}\} + S + \text{块}, 0, 1)$;
5. $\{\text{repeat}\} + S + \text{块} + S + \{\text{while}\} + S + \text{条件} + E + \{?\}$;
6. $\{\text{while}\} + S + \text{条件} + S + \text{块}$;
7. $\{\text{return}\} + W + \text{表达式} + E + \{?\}$;
8. $\{!\}$ 。

代码块 是以下之一：

1. 声明 $+ E + \{(\} + E + \{)\} + S + \text{块}$;
2. 声明 $+ E + \{(\} + E + \text{声明} + E + r(\{,\} + S + \text{声明}, 0) + E + \{)\} + S + \text{块}$;
3. $\{\text{main}\} + E + \{(\} + E + \{)\} + S + \text{块}$ 。

程序 是 注释 $+ r(L + \text{代码块}, 1) + E$ 。

程序、代码块、陈述、条件、块、表达式和声明都是 非终结符。其它符号（标识符, E , $\{?\}$ 等）是 终结符。现在我们可以通过下面的方式生成一份代码：

1. 新建一个列表，初始值为 [程序]。
2. 每次将列表中第一个非终结符根据上文中的规则进行替换，直到列表中不存在非终结符。（例如，一个 [声明] 可以被替换为 [标识符, W , 标识符]）
3. 将列表中的每一个元素用其对应集合中的一个字符串取代掉。

将这个列表中的所有字符串首尾相接，就可以得到一份代码。这个列表叫做这份代码的分解。可以证明，任意一份代码的分解都是唯一的。

两份代码相等，当且仅当它们的分解的长度相等，同时，对于它们的分解中位置对应的一对字符串，要么相等，要么都不包含空格、制表符、换行符以外的任何字符。

当 $S = W = \{\text{空格}\}$, $L = \{\text{换行符}\}$, $E = \{\epsilon\}$ 时，对应的程序被称为 半格式化程序。

一份代码是 格式化良好的 当且仅当它可以通过在一份属于半格式化程序的代码中的每个换行符（ASCII 码为 10）后插入 $k \times x$ 个空格，其中 x 表示该换行符所处块的块的数量（特别地，假如这个换行符是某个块中的最后一个换行符，这个块将不会计入 x 中）， k 是一个给定的数。

给定一份代码，找到一份格式化良好的且与给定代码相等的代码。

Input

第一行一个非负整数 k 。

第二行开始一直到输入的末尾，是原本的代码。

保证输入长度不超过 10^5 字节。

Output

输出一份与输入代码相等的**格式化良好的**的代码。注意，不允许输出多余的空格、制表符、或换行符。

保证输出长度不超过 2×10^5 字节。

Note

如果你需要阅读样例，请访问竞赛网站。

Problem B. 修整回文串 (trim)

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 600 megabytes

紫眼是一个很可爱的女孩子！她对事物的看待也非常独特！她最喜欢的就是回文字符串。

她定义了两个回文字符串的一个值 $g(s, t)$ (可以为零) 表示满足 u 是 s 的后缀且 t 是 u 的后缀的回文字符串 u 的个数。

紫眼有一个长度为 n 的回文字符串 s 。她有一天闲来无事，在 s 里选了一个回文子串作为字符串 t 。可是这个字符串太难看了！她决定开始修整这个串。

她可以对字符串修整任意次 (可以为零次)：每次修整是指移除字符串的一个非空前缀，留下一个回文后缀 (可以为空)。但是她无法修整一个空串，也就是说只要字符串变成了空串，她就只得停止修整。

给定整数 m 和三个数组 $a_1, a_2, a_3, \dots, a_m, b_1, b_2, b_3, \dots, b_m, c_0, c_1, c_2, \dots, c_n$ 。每次修整都会使紫眼的心情变化。具体地，当她把字符串从 U 修整成了 V 时，她能获得 $a_{\min(z, m)} \cdot g(U, V) + b_{\min(z, m)} c_{|V|}$ 的愉悦度，其中 z 表示满足 s' 是 s 的回文子串，且 V 是 s' 的后缀的不同字符串 s' 的个数。

请注意愉悦度有可能是负数。

紫眼发现自己能收获很多愉悦度！现在她想知道她如果按照最优的办法选取字符串 t 并对它修整，能获得最大的愉悦度是多少。

Input

输入的第一行包括两个整数 n, m ($1 \leq n < m \leq 10^6$)。

输入的第二行包括一个回文字符串 s ($|s| = n$)，其中 s 仅包含小写拉丁字母。

输入的第三行包含 m 个整数表示数组 a_1, a_2, \dots, a_m ($|a_i| \leq 2 \times 10^5$)，满足 $a_i \leq a_{i+1}$ 。

输入的第四行包含 m 个整数表示数组 b_1, b_2, \dots, b_m ($|b_i| \leq 2 \times 10^5$)。

输入的第五行包含 $(n+1)$ 个整数表示数组 c_0, c_1, \dots, c_n ($|c_i| \leq 2 \times 10^5$)。

Output

一行一个整数表示最大愉悦度。

Examples

standard input	standard output
3 4 efe -6 2 3 8 -3 -7 9 -2 9 -9 -1 4	67
10 11 nopqrrqpon -989 -955 -878 -857 -826 -100 368 582 636 759 762 914 73 397 649 320 -621 -280 -321 -253 294 157 884 -397 -623 970 794 -724 223 -233 -74 629 488	141074

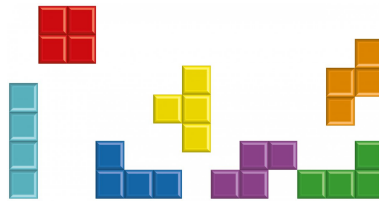
Problem C. 哈克萨吉斯 (hexagis)

Input file: standard input
Output file: standard output
Time limit: 6 seconds
Memory limit: 256 megabytes

在问题开始之前, 先给出一些定义。

- 在方格表中, 称两个格子 **相邻** 当且仅当它们存在各自的一条边恰好重合。
- 在方格表中, 称两个格子 **可达** 当且仅当存在一个移动方案使得起点为其中一个格子, 终点为另一个格子, 且每次移动只移动至与当前格子相邻的格子中。
- 在方格表中, 称一个格子子集 **四连通** 当且仅当去除不在子集中的其他格子后, 子集中的格子两两可达。此时这个子集被称作 **区域**。

在著名游戏"俄罗斯方块"中, 一共有七种不同的 **四连块**, 它们是所有可能的四连通的四个格子的形状, 其中通过旋转和平移能够互相重合的算作同一种形状。



"俄罗斯方块"中的七种四连块

然而, 你觉得"俄罗斯方块"非常简单。这是因为其中四连块的种数太少了, 导致它们能产生的组合比较少, 让这个游戏变得很简单。于是, 你决定游玩一个类似的却更难的游戏: "哈克萨吉斯"。

游戏"哈克萨吉斯"中的 **六连块** 有 60 种, 它们是所有可能的四连通的六个格子的形状, 其中通过旋转和平移能够互相重合的算作同一种形状。

你需要做的, 就是使用每一种六连块恰好一个, 将它们不重不漏地放进一个 19×19 的方格表中, 并且除了方格表正中间的格子以外, 其它格子恰好被一个六连块覆盖。

这个游戏的难度显著提高, 你抓耳挠腮却无法找到一个正确的解答。所以你决定和你的朋友一起玩。然而, 你的朋友声称他把这个难题秒了, 并且为了证明自己, 他给出了 T 个他的解答。不过, 通过找到每一种六连块的使用次数, 你发现他的解答全都是错误的。

他现在转而认为这个游戏无解了。你能找到一个解答来证明他是错的吗?

Input

输入包含 $19T + 1$ 行:

- 第一行包含一个正整数 T ($1 \leq T \leq 500$) 表示朋友解答的数量。
- 接下来 $19T$ 行, 第 $19k + i - 18$ ($1 \leq k \leq T, 1 \leq i \leq 19$) 行包含一个长度为 19 的字符串 $S_{k,i}$ 表示第 k 个解答的方格表第 i 行的信息。

除去字符串 $S_{k,10}$ ($1 \leq k \leq T$) 中第 10 个字符为 ':' 外, 其余字符均为阿拉伯数字。四连通的相同数字代表同一个区域。

Output

输出包含 $T + 19$ 行:

1. 若第 k ($1 \leq k \leq T$) 个解答中含有不是六连块的区域, 在第 k 行输出 -1 , 否则在第 k 行输出 60 个非负整数 $a_{k,1}, a_{k,2}, a_{k,3}, \dots, a_{k,60}$, 表示在 **单调不增** 的顺序下所有六连块的使用次数。
2. 接下来 19 行, 第 $T+i$ ($1 \leq i \leq 19$) 行包含一个长度为 19 的字符串 R_i 表示正确解答的方格表第 i 行的信息。

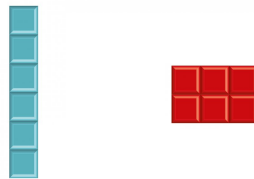
你需要保证除去字符串 R_{10} 中第 10 个字符为 ‘.’ 外, 其余字符均为阿拉伯数字。四连通的相同数字代表同一个区域。不满足格式要求或不满足题意的输出将会被视为答案错误。

若有不止一个正确答案, 输出任意一个即可。

Note

如果你需要阅读样例, 请访问[竞赛网站](#)。

朋友的第一个解答使用了 32 个左侧的六连块与 28 个右侧的六连块, 如下图所示。



在朋友的第二个解答中, 出现了不是六连块的区域。

在输出正确解答的部分中, 样例输出 **不** 满足题意。故样例输出将会被视为答案错误。

Problem D. 语法测试 (grammar)

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

有 n 个布尔变量组成数列 $a_1, a_2, a_3, \dots, a_n$ 。对于初值为任意非负整数的变量 x, y ，按照整数 i 从 $n-1$ 到 1 的顺序，分情况执行以下操作 (\oplus 表示二进制按位异或)：

若 a_i 为	真	真	假	假
若 a_{i+1} 为	真	假	真	假
执行	$x := x \oplus x$	$x := x \oplus y$	$y := y \oplus x$	$y := y \oplus y$

统计不同数列 $a_1, a_2, a_3, \dots, a_n$ 的数量，满足在该数列上，对初值为任意非负整数的变量 x, y 执行上述所有操作，结果总是使这两个变量交换它们的初始值（也就是说， x 的终值是 y 的初值， y 的终值是 x 的初值）。答案对 998244353 取模。

Input

本题包含多组测试数据。

首先在第一行输入一个正整数 T ($1 \leq T \leq 10$) 表示测试数据组数。

对于每一组测试数据：

输入的唯一一行包含一个正整数 n ($1 \leq n \leq 10^{18}$) 表示数列的预期长度。

Output

对于每一组测试数据，输出包含一行一个非负整数表示满足条件的数列数量对 998244353 取模后的值。

Example

standard input	standard output
3	0
1	0
2	2
4	

Problem E. 老师与好感度 (affection)

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

现在有 n 位学生, 编号为 $1, 2, 3, \dots, n$, 初始 好感度 依次为 $a_1, a_2, a_3, \dots, a_n$ 。老师在每一天可以执行以下操作恰好一次:

- 选取整数 l, r 满足 $1 \leq l \leq r \leq n$, 令所有编号属于 $[l, r]$ 的学生的好感度增加 1。具体地, 对于所有整数 i ($l \leq i \leq r$), 令 $a_i := a_i + 1$ 。

现在, 老师希望通过尽量少的天数使得学生们的好感度数列 $a_1, a_2, a_3, \dots, a_n$ 中至多出现 m 种不同的数。他找来了你帮忙求出天数的最小值。

Input

本题包含多组测试数据。

首先在第一行输入一个正整数 T ($1 \leq T \leq 10$) 表示测试数据组数。

对于每一组测试数据:

第一行包含两个整数 n, m ($1 \leq n \leq 400, 1 \leq m \leq 2, n \geq m$), 表示学生的数量和求解答案所需的参数。

第二行包含 n 个非负整数 $a_1, a_2, a_3, \dots, a_n$ ($0 \leq a_i \leq 100$), 表示每个学生的初始好感度。

Output

对于每一组测试数据, 输出包含一行一个非负整数表示天数的最小值。

Example

standard input	standard output
4	2
3 1	12
3 1 3	3
5 1	1
5 1 6 1 8	
5 2	
5 1 6 1 8	
5 2	
1 3 2 3 1	

Problem F. 老师和 Yuuka 逛商场 (yuuka)

Input file: standard input
Output file: standard output
Time limit: 3 seconds
Memory limit: 256 megabytes

老师最近出了一些程序设计竞赛试题，赚到不少钱。除去购买草莓牛奶所用的一部分，还留有许多剩余。Yuuka 是今天的值日生，凑巧今天工作又不多，老师决定带她去商场逛逛，以弥补曾经对她开的关于体重的玩笑。

在购买商品之前，老师已得知商场中有 n 个排成一排的商店。从左到右这些商店编号为 $1, 2, 3, \dots, n$ 。Yuuka 列出了每个商店的**独特度**：商店 i ($1 \leq i \leq n$) 的独特度为正整数 a_i 。

现在，老师需要将所有商店划分成三个连续的部分，从左到右编号为 $1, 2, 3$ 。

具体地，老师会指定两个整数 b_1, b_2 ($1 < b_1 < b_2 \leq n$)。令 $b_0 = 1, b_3 = n + 1$ ，此时 b_1, b_2 对应了一个满足下列条件的划分方案：

- 第 i ($1 \leq i \leq 3$) 个部分包含商店 $b_{i-1}, b_{i-1} + 1, b_{i-1} + 2, \dots, b_i - 1$ 。

对于每一个独特度 x ，假如三个部分中每个部分都存在至少一个独特度为 x 的商店，则 Yuuka 称独特度 x 是**普遍的**。

老师从不是一个小气的人，但他还是想将这些钱用在更重要的事情上。为了降低 Yuuka 对商品的消费，老师想要通过合适的划分手段，使得这个划分中普遍的**独特度的个数**最大。他找来了你帮忙，并且要求你额外提供一个达到这个最大值的划分方案。

Input

本题包含多组测试数据。

首先在第一行输入一个正整数 T ($1 \leq T \leq 10$) 表示测试数据组数。

对于每一组测试数据：

第一行包含一个正整数 n ($3 \leq n \leq 150,000$)，表示商店的数量。

第二行包含 n 个正整数 $a_1, a_2, a_3, \dots, a_n$ ($1 \leq a_i \leq 10^6$)，表示每个商店的独特度。

Output

对于每一组测试数据，输出包含两行：

第一行包含一个整数表示题目所求的最大值。

第二行包含两个整数 b_1, b_2 。你需要保证 $1 < b_1 < b_2 \leq n$ 。

若有多个满足条件的划分方案，输出任意一个都算作正确。

Example

standard input	standard output
3	3
9	4 7
1 2 3 3 2 1 2 3 1	1
10	2 3
3 3 2 2 2 3 2 1 1 1	2
10	3 6
4 3 3 3 4 4 3 4 3 3	

Note

来自 Yuuka 的温馨提示：本题输入量较大，请选择合适的输入方式。

这个样例共有三组测试数据。

在第一组测试数据中, 令 $b_1 = 4, b_2 = 7$ 。此时独特度 1, 2, 3 是普遍的, 普遍的独特度的个数达到最大值。

在第二组测试数据中, 令 $b_1 = 2, b_2 = 3$ 。此时独特度 3 是普遍的, 普遍的独特度的个数达到最大值。值得一提的是, 在这组测试数据中, 存在不止一个划分方案达到题目所求最大值, 输出任意一个都算作正确。

在第三组测试数据中, 令 $b_1 = 3, b_2 = 6$ 。此时独特度 3, 4 是普遍的, 普遍的独特度的个数达到最大值。这组测试数据同样存在不止一个划分方案达到题目所求最大值, 输出任意一个都算作正确。

Problem G. 星光 (starlight)

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 32 megabytes

给定长度为 n 的序列 a 和正整数 m 。你需要维护 q 次操作。
每次操作给定 x, y, z , 表示:

1. 首先, 永久地将 a_x 修改为 y 。
2. 接下来, 询问最少需要进行多少次删除, 才能使得序列中存在至少 z 个数为 0。

定义一次 **删除** 指:

1. 将序列中的 **非零** 元素按照大小从小到大排序, 假设排序后这些元素的 **下标** 序列为 $b_1, b_2, b_3, \dots, b_k$ 。
2. 任选一个不大于 m 的非负整数 k' 。对所有满足 $1 \leq i \leq \min(k, k')$ 的 i , 令 $a_{b_i} := a_{b_i} - 1$ 。

注意:

- 修改是 **永久生效** 的, 影响后续询问。
- 每次询问后序列将恢复原状, 也就是说删除 **不永久生效**。

Input

第一行三个正整数 n, m, q ($1 \leq n, m, q \leq 10^5$), 含义见题目描述。
第二行 n 个正整数表示序列 a ($1 \leq a_i \leq 10^9$)。
接下来 q 行, 每行三个整数 x', y', z' , 表示一次操作。这次操作中, $x = 1 + ((x' + \text{lans}) \bmod n)$, $y = 1 + ((y' + \text{lans}) \bmod 10^9)$, 以及 $z = 1 + ((z' + \text{lans}) \bmod n)$ 。
在这里, lans 表示上一次操作的答案 (如果不存在上一次操作, 则 $\text{lans} = 0$)。保证 $x', y', z' \in [0, 2^{63})$ 。

Output

对于每次操作, 输出一行一个整数表示答案。

Example

standard input	standard output
5 4 5	400488027
793982278 930973954 65057664 63383158	857365436
226369572	857365436
1815647561 1400488026 3187009203	1056063199
968072559 3087832345 1063159872	226369572
1111136800 2762585403 1385851423	
759541154 2135314604 2835883033	
880295667 717958068 387325293	

Problem H. 合法等式等法合 (equation)

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

定义:

- 一个 **合法算式** 为一个 **非负一位数** 或者中间被一个运算符 (+、-、* 中的一个) 隔开的两个 **非负一位数**。
- 一个合法算式的 **值** 为其在数学上运算得到的结果, 如 3 的值为 3、3-4 的值为 -1、5*6 的值为 30。
- 一个 **合法等式** 为一系列 (至少两个) 的值相等的合法算式, 由 = 隔开。如 3=3、1+2=3、3-2=1-0=1 均是合法等式, 而 3+=5、8+8=16、=5+3=8、-1+5=4 均不是。

紫眼有一个长度为 4 的字符串 s , 满足 s_0 与 s_2 为一位数, s_1 为 +、-、* 中的一个, s_3 为 =。

现在紫眼想要在 s 中的任意位置 (包括开头与结尾) 插入一些字符以得到一个 **回文** 字符串。由于她很懒, 因此只有那些通过添加数量最少的字符得到的回文字符串才会被考虑。

她想知道是否存在一个回文串被考虑到, 且可以看作一个 **合法等式**。

由于紫眼一直认为大于七的质数只有一个, 而且可怜的她为了寻找那个质数已经三天没睡觉了, 所以这个问题就只好交给你了。

Input

本题包含多组测试数据。

第一行包含一个整数 T ($1 \leq T \leq 300$) 表示测试数据组数。

对于每一组测试数据:

输入的唯一一行包含一个长度为 4 的字符串 s ($0 \leq s_0, s_2 \leq 9$, s_1 为 +、-、* 中的一个; s_3 为 =)。

Output

对于每组测试数据, 如果存在一个合法的字符串, 输出 Yes; 否则输出 No。

Example

standard input	standard output
3	Yes
1+2=	No
1*1=	No
5-3=	

Note

对于第一组测试数据, 由于字符串 1+2=2+1 是一个被考虑到的合法等式, 因此答案是 Yes。

对于第二组测试数据, 只有字符串 =1*1= 会被考虑到。答案是 No。

Problem I. 矩阵 (matrix)

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

Piggy 有一个 $n \times m$ 的 **循环** 矩阵。这意味着第 n 行与第 1 行是相邻的, 且第 m 列与第 1 列是相邻的。你需要用 1 至 nm 的整数填充这个矩阵使得它满足以下条件:

- 每个 1 至 nm (闭区间) 的整数都在矩阵中出现了恰好一次。也就是说, 恰好一个矩阵中的格子包含了每个整数。
- 对于每个 $1 \leq i \leq nm - 1$, 包含 i 的格子与包含 $i + 1$ 的格子要么在同一行, 要么在同一列。
- 对于每个 $1 \leq i \leq nm - 1$, 如果 Piggy 初始在包含 i 的格子处, 那么存在左、右、上和下中的一个方向, 使得若 Piggy 朝着那个方向移动恰好 i 格, 他可以到达包含 $i + 1$ 的格子处。注意矩阵是循环的, 这意味着如果他在第 n 行向下移动一格, 会移动到第 1 行, 如果在第 m 列向右移动一格, 会移动到第 1 列, 反之亦然。

构造一个合法的矩阵, 或者报告不存在合法的矩阵。

Input

一行包含两个正整数 n, m ($1 \leq n, m, nm \leq 10^6$)。

Output

第一行输出 YES 或 NO, 表示你是否找到了一个合法的矩阵。

如果答案是 YES, 接下来输出 n 行, 每行包含 m 个整数, 描述你构造的合法矩阵。

若有多个合法矩阵, 输出任意一个都算作正确。

Examples

standard input	standard output
4 5	YES 1 9 14 15 8 2 10 3 16 17 20 19 4 5 18 12 11 13 6 7
4 6	NO

Problem J. 线段 (segments)

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

小 A 正在上文化课, 很可惜他什么都听不懂。这时, 他发现了笔盒中的笔长度都相同。于是他把它们拿出来开始玩。

由于这是真实事件, 所以本题发生在 **三维空间** 中。

小 A 想要确定空间中的 $n + 2$ 个实数坐标点 $V_1 = (x_1, y_1, z_1), V_2 = (x_2, y_2, z_2), V_3 = (x_3, y_3, z_3), \dots, V_{n+2} = (x_{n+2}, y_{n+2}, z_{n+2})$ (点可以重合) 使得它们满足以下要求:

1. $x_{n+1} = x_1, y_{n+1} = y_1, z_{n+1} = z_1$ 。
2. $x_{n+2} = x_2, y_{n+2} = y_2, z_{n+2} = z_2$ 。
3. 对于所有整数 i ($1 \leq i \leq n$), $\text{dis}(V_i, V_{i+1}) = 1$, 其中 $\text{dis}(A, B)$ 表示实数坐标点 A, B 在空间中的欧几里得距离。
4. 对于所有整数 i ($1 \leq i \leq n$), 线段 $V_i V_{i+1}$ 垂直于线段 $V_{i+1} V_{i+2}$ 。

小 A 需要你给出一组满足要求的方案中 $V_1, V_2, V_3, \dots, V_n$ 的坐标。若有多组满足要求的方案, 输出任意一组即可。若不存在满足要求的方案, 则不需要输出。

同时, 他还需要你给出满足要求的不同方案数量。若有无穷多组满足要求的方案, 则输出 **inf**。若不存在满足要求的方案, 则输出 **-1**。

定义两个方案不同当且仅当它们无法通过平移、旋转与轴对称的任意组合使得 n 个点重合。

Input

第一行一个整数 T ($1 \leq T \leq 10^5$), 表示测试用例数量。

第 2 行到第 $t + 1$ 行每行输入一个整数 n ($2 \leq n \leq \sum n \leq 10^6$), 表示点的数量。

Output

对于每一个测试用例:

若存在满足要求的方案, 则首先输出 n 行, 第 i ($1 \leq i \leq n$) 行包含三个实数分别表示 x_i, y_i, z_i 。接下来在第 $n + 1$ 行输出一个整数或一个字符串表示满足要求的不同方案数量。若有无穷多组满足要求的方案, 输出 **inf**。

若不存在满足要求的方案, 则输出一行一个 **-1**。

本题启用 Special Judge。你输出的方案需要满足对于所有整数 i ($1 \leq i \leq n$), $|\text{dis}(V_i, V_{i+1}) - 1| \leq 10^{-6}$, 且 $|\overrightarrow{V_i V_{i+1}} \cdot \overrightarrow{V_{i+1} V_{i+2}}| \leq 10^{-6}$ 。

Example

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

Problem K. 神奇集合 (amazing)

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

给定一棵有根树 T , 包含 n 个顶点和 $n - 1$ 条有向边。

对于顶点 i , 其权值为 a_i 。如果存在一条从顶点 y 出发、到顶点 x 结束的路径, 且该路径仅通过有向边, 则称顶点 x 可从顶点 y **到达**。保证所有顶点都可以从根节点到达。

给定 m 对顶点 (u_i, v_i) , 满足在原树中 u_i 可从 v_i 到达。现在, 将添加新的有向边 $u_i \rightarrow v_i$ 以形成一个新的图。

定义一个顶点集合的子集 S (可以为空) 为 **神奇** 的, 当且仅当对于 S 中的每个元素 x , 在新图中所有可从 x 到达的顶点都在 S 中。**神奇** 集合 S 的权值定义为 S 中所有元素的权值之和。空集的权值为 0。

对于所有 **神奇** 集合, 共有多少种不同的权值出现?

Input

第一行包含一个整数 n ($1 \leq n \leq 10^4$), 表示树 T 的顶点数量。

第二行包含 n 个整数, 第 i 个数表示 a_i ($0 \leq a_i \leq \sum_{i=1}^n a_i \leq 10^4$)。

接下来的 $n - 1$ 行描述树的所有 $n - 1$ 条有向边。对于第 i 行, 给出两个整数 U_i, V_i ($1 \leq U_i, V_i \leq n$), 表示存在一条从 U_i 指向 V_i 的边。

接下来一行包含一个整数 m ($1 \leq m \leq 5 \times 10^5$), 表示新增边的数量。

接下来的 m 行描述所有新增的有向边。对于第 i 行, 给出两个整数 u_i, v_i ($1 \leq u_i, v_i \leq n$), 表示新增一条从 u_i 指向 v_i 的边。

Output

一个整数, 表示上述问题的答案。

Example

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

Note

样例包含三个 **神奇** 集合: $\{1, 2, 3\}$ 、 $\{3\}$ 、 \emptyset 。由于它们的权值互不相同, 答案为 3。

新增的边 (u_i, v_i) 不保证两两不同。

新增的边 不保证 $u_i \neq v_i$ 。

树的根节点 未 直接给出。

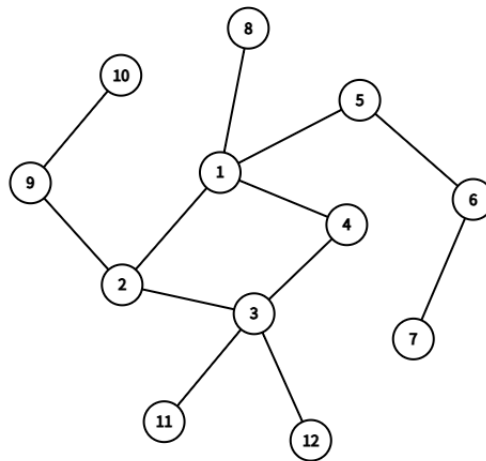
Problem L. 沙族馆 (desertrium)

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

小 ppip 正在梦中图寻。梦里他得到了一个沙漠图。由于他的梦很抽象，因此他在图中找到了许多水产。他想知道这个图中水产的数量，并找你帮忙。

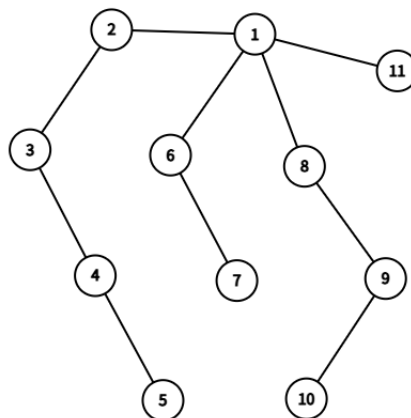
具体地，你需要计算子水母、子海星和子鳗鱼的数量。定义如下。

- 一个 **沙漠** 由至少一个仙人掌构成。
- 一个 **仙人掌** 是一张简单无向连通图，满足每个点至多处在一个长度不小于 3 的简单环中。
- 一个 **水母** 是一张简单无向连通图，满足它恰好包含一个长度不小于 3 的简单环，且所有不在环上的点的度数不超过 2。形象地说，它由多条链挂在一个环上构成。作为例子，以下是一个水母：



简单环是 $1 - 2 - 3 - 4$

- 一个大小为 k 的 **子水母** 是一个二元组 (V, E) ，其中 V 是大小为 k 的点集， E 是大小为 k 的边集，满足 (V, E) 是一个水母。两个子水母不同当且仅当它们的边集不同。
- 一个 **海星** 是一张简单无向连通图，满足它不包含任何简单环（也就是一棵树），且有至多一个点的度数超过 2。形象地说，它由多条链挂在一个点上构成。作为例子，以下是一个海星：



点 1 的度数为 4

- 一个大小为 k 的 **子海星** 是一个二元组 (V, E) , 其中 V 是大小为 k 的点集, E 是大小为 $k - 1$ 的边集, 满足 (V, E) 是一个海星。两个子海星不同当且仅当它们的边集不同。
- 一个 **鳗鱼** 是一条链, 也就是一张简单无向连通图, 满足它不包含任何简单环, 且不存在任何点的度数超过 2。
- 一个大小为 k 的 **子鳗鱼** 是一个二元组 (V, E) , 其中 V 是大小为 k 的点集, E 是大小为 $k - 1$ 的边集, 满足 (V, E) 是一个鳗鱼。两个子鳗鱼不同当且仅当它们的边集不同。

值得一提的是, 有些子海星也同样是子鳗鱼, 它应当在答案中被统计两次。额外地, 任何上述列举出的水产中点的数量必须不少于 2。
由于答案可能很大, 将它对 998244353 取模。

Input

本题的输入含有多组测试数据。

第一行一个正整数 T ($1 \leq T \leq 10^6$), 表示测试数据的组数。

对于每组测试数据:

第一行两个整数 n, m ($1 \leq n \leq 10^5, 0 \leq m \leq 2n, 1 \leq \sum n \leq 10^6$), 表示沙漠的顶点数和边数。

接下来, 第 $i + 1$ ($1 \leq i \leq m$) 行有两个整数 u_i, v_i ($1 \leq u_i, v_i \leq n, u_i \neq v_i$), 表示沙漠中的一条边。保证所有的无序对 (u_i, v_i) 两两不同, 即沙漠中没有重边。

Output

对于每组测试数据, 输出一行一个整数, 表示沙漠中水产的总数对 998244353 取模后的结果。

Example

standard input	standard output
1 10 10 1 4 2 8 5 9 8 4 7 9 5 3 6 1 9 3 10 3 2 4	83