

Problem A. 2026美加墨世界杯

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

2024年6月6日晚，在沈阳进行的世界杯亚洲区预选赛第二阶段第五轮的一场比赛，中国男足主场1-1战平泰国，拿到宝贵的一分。

第二阶段，中国队被分在C组，同组对手有韩国、泰国和新加坡队。五轮比赛过后，中国队以2胜2平1负的成绩积8分暂列小组第二，泰国积5分排名第三。

今天的这场比赛对两支球队而言都非常关键。中国队需要一场胜利就能提前一轮从小组出线。相反，若泰国队拿到分数，最后一轮他们将对阵同组实力最弱的新加坡队，而将在客场挑战韩国队的中国队将面临巨大压力，形势非常被动。

排兵布阵方面，中国队主帅伊万科维奇也面临着"无米之炊"的尴尬局面，当家射手武磊和中场主力李源一停赛，张玉宁因伤也无法首发登场。

最终，对比上一场4-1击败新加坡的比赛，中国队本场做出五处调整，王振澳顶替高准翼，王上源、徐皓阳顶替李源一、谢鹏飞，前场韦世豪、阿兰顶替武磊和张玉宁，其中王振澳和徐皓阳均迎来国家队首秀：14-王大雷；19-刘洋、2-蒋光太、5-朱辰杰、4-王振澳；6-王上源、7-徐皓阳、8-高天意、17-费南多；11-韦世豪、22-阿兰。

今天这场比赛对双方来说都是输不起的一场比赛，所以比赛初期场上节奏飞快，攻防转换就在电光火石之间。

第11分钟，泰国队发动快速反击，左路创造出绝佳机会，韦世豪一路狂奔到禁区内化解险情。12分钟，泰国队开出角球，素帕南禁区外围张弓搭箭，打出一记质量极高的远射，可惜稍稍偏出。

第15分钟，阿兰右脚扣球左脚吊传入禁区，韦世豪左路停球直接射门，无奈角度太小偏出球门。

第20分钟，泰国队后场长传直接找到前场左侧的素巴猜，他左路一对一晃过蒋光太杀到禁区左侧底线后传中，中路素巴楚跟上鱼跃冲顶破门。0-1！中国队主场陷入被动。

失球后中国队立刻还以颜色，阿兰禁区前沿假动作闪出角度后地平球射门，皮球打在立柱上弹出，费南多抢到第二点补射，被对方后卫在门线上解围。

上半场最后一分钟，中国队又逃过一劫。素巴楚禁区右侧一个近乎单刀的机会选择地平球推射，皮球越过王大雷打在立柱上。

纵观上半场，中国队在攻防两端都落了下风。尤其是防守端，没有给泰国队足够的压迫感，所以对方能够轻易在两边路拿球发动快速反击，最终导致丢球。

易边再战，第49分钟泰国队后卫禁区内手球送出点球，中国球员费南多12码点射力量稍大，虽骗过门将但皮球还是飞出横梁，中国队错过了扳平比分的绝佳机会。

随后的比赛，落后的中国队对泰国队形成围攻之势，但难以创造出绝对机会。而泰国队收缩阵型将球权交给中国队，伺机发动快速反击。

第69分钟，费南多前场抢断晃过对方门将，无奈角度太小无法打门，仓促之下传中，泰国后卫将球解围。

第76分钟，中国做出换人调整，16号蒋圣龙和9号张玉宁临危受命，换下阿兰和王上源。

第78分钟，中国队利用任意球机会扳平比分。谢鹏飞任意球吊入小禁区前沿，朱辰杰抢到第一点摆渡，禁区里右侧拜合拉木跟上凌空抽射将比分扳平！1-1，小将拜合拉木首次代表国家队就收获了处子球。

扳平比分之后，中国队攻势如潮，不断从两边路起球，充分利用身高优势来制造威胁，张玉宁门前连续两次险些破门。最后的时间里，中国队得势不得分，场上比分再未改写。

赛后，赛事积分榜如下：

队伍	总场数	胜场数	平场数	负场数	总进球数	总失球数	总净胜球数	积分
韩国	5	4	1	0	19	1	18	13
中国	5	2	2	1	9	8	1	8
泰国	5	1	2	2	6	8	-2	5
新加坡	5	0	1	4	4	21	-17	1

其中，相关参数的具体计算方式如下：

- 积分的计算规则是：胜一场获胜队伍积 3 分，平一场双方各积 1 分，负一场败方不积分。也就是说 $\text{积分} = 3 \times \text{胜场数} + 1 \times \text{平场数}$ 。
- 总进球数为队伍在各场比赛中的进球数之和。
- 总失球数为队伍在各场比赛中的失球数之和。
- 总净胜球数为队伍的总进球数减去总失球数。

每个小组分别有四只球队，排名最高的两支球队将晋级下一阶段的比赛，其排名按照以下规则确定：

1. 首先比较积分，积分高者排名靠前。
2. 积分相同的话，再比较总净胜球数，总净胜球数高者排名靠前。
3. 总净胜球数也相同的，再比较总进球数，总进球数高者排名靠前。
4. 如果按照以上规则，仍有两支或两支以上的球队并列，则按以下规则确定排名先后：
5. 仅考虑并列球队之间进行的比赛，计算并列的各支队伍在这几场比赛中的积分、总净胜球数以及总进球数。
6. 首先比较按照第 5 条规则计算出来的积分，积分高者排名靠前。
7. 如果积分相同，再比较按照第 5 条规则计算出来的总净胜球数，总净胜球数高者排名靠前。
8. 如果总净胜球数也相同，再比较按照第 5 条规则计算出来的总进球数，总进球数高者排名靠前。
9. 如果按照以上规则，依旧有两支或两支以上的球队并列，这样就需要比较公平竞赛积分。这就不在本题的考虑范围之内了。

最后一轮比赛，中国队对阵韩国队，泰国队对阵新加坡队，而这两场比赛的结果决定中国队能否晋级下一阶段。假设最后一轮的结果如下：

- 韩国 $a : b$ 中国；
- 泰国 $c : d$ 新加坡。

并且，在此之前，中国队在客场 2 : 1 战胜泰国队，在主场 1 : 1 平泰国队，也就是说，中国队相互比赛的积分是高于泰国队的（参照规则第 5 条）。小Z想知道，在上述条件下，中国队能否晋级世界杯预选赛下一阶段。

可以证明，在本题的数据范围内，按照上述规则一定可以完成排名的计算，无需涉及到比较公平竞赛积分这一步。

Input

仅一行，包含四个整数 a, b, c, d ($0 \leq a, b, c, d \leq 10$)，用空格分隔，含义如题面所述。

Output

输出一行，表示题目所求的结果。

若中国队能够进入世界杯预选赛下一轮，输出 **YES**。否则，输出 **NO**。

你可以以任意的大小写形式输出结果。也就是说，**yES**、**yes**以及**Yes**都会被认为是**YES**；**no**以及**No**都会被认为是**NO**。

Example

standard input	standard output
1 0 3 1	YES

Note

若比赛结果如样例所示，即：

- 韩国 1 : 0 中国；
- 泰国 3 : 1 新加坡。

比赛结束后的赛事积分榜如下：

队伍	总场数	胜场数	平场数	负场数	总进球数	总失球数	总净胜球数	积分
韩国	6	5	1	0	20	1	19	16
中国	6	2	2	2	9	9	0	8
泰国	6	2	2	2	9	9	0	8
新加坡	6	0	1	5	5	23	-19	1

根据赛事规则：

1. 中国队与泰国队积分相同。
2. 中国队与泰国队总净胜球数相同。
3. 中国队与泰国队总进球数相同。
4. 在中国队与泰国队相互之间的比赛中，中国队获得的积分高于泰国队。

中国队成功以小组第二晋级世界杯预选赛下一阶段！

Problem B. Stop! High School Maths Please No More

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

Rubyonly有一个长度为 n 的正整数数列 $[a_1, a_2, a_3, \dots, a_n]$ 。

Rubyonly喜欢等比数列，因此，他打算对这个数列进行若干次修改。

每次修改，Rubyonly都将选择一对正整数 i 和 x ($1 \leq i \leq n, 1 \leq x$)，并将 a_i 修改为 x 。注意，这里的 x 可以是任意大的有限正整数。

Rubyonly希望通过尽可能少的修改次数，将原数列修改成一个公比为正整数的等比数列。

更具体的讲，我们记修改后的数列为 $[b_1, b_2, b_3, \dots, b_n]$ ，需要存在一个正整数 q ，使得对于任意的 $1 \leq i \leq n - 1$ ，都有 $b_{i+1} = q \cdot b_i$ 成立。

请你帮Rubyonly计算一下，他至少需要进行几次修改，才能达成这一目标。

Input

输入的第一行包含 1 个正整数 n ($2 \leq n \leq 10^5$)，表示原数列的长度。

输入的第二行包含 n 个正整数 $a_1, a_2, a_3, \dots, a_n$ ($1 \leq a_i \leq 10^{18}$)，用空格分隔，表示原数列中的数。

Output

输出一行，包含 1 个整数，表示最少需要的修改次数。

Examples

standard input	standard output
5 1 3 4 5 16	2
7 1 2 1 2 1 2 1	3
2 2 1	1
4 1 2 16777216 140737488355328	2
3 1 1000000000 999999999999999999	1
3 1 4294967297 8589934593	1
5 1 4 6 8 10	3
5 1 100 10000 1000000 100000000	0

Note

在样例一中，可以通过 2 次修改，将原数列修改为 $[1, 2, 4, 8, 16]$ 。

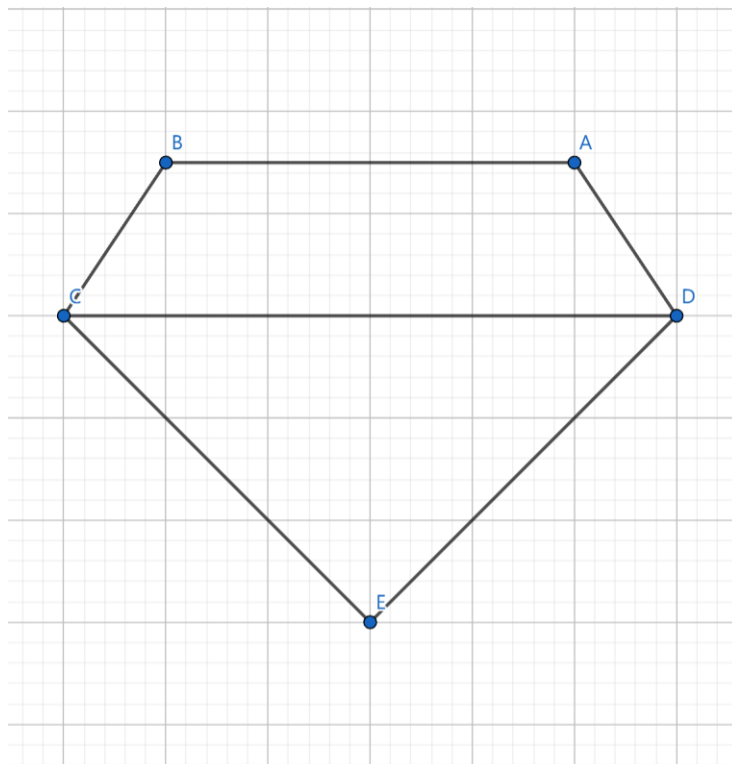
在样例二中，可以通过 3 次修改，将原数列修改为 $[1, 1, 1, 1, 1, 1, 1]$ 。

Problem C. Diamond

Input file: standard input
Output file: standard output
Time limit: 3 seconds
Memory limit: 512 megabytes

RubyOnly 很喜欢钻石。

在一个二维平面内，一个钻石由五个点 A, B, C, D, E 构成，如下图所示：



具体地，在二维平面中，五个点 A, B, C, D, E 形成一个钻石形图案当且仅当五元组 (A, B, C, D, E) 满足以下所有条件：

- A, B, C, D, E 五个点互不相等。
- $|BC| = |AD|$, $|EC| = |ED|$, $|BA| < |CD|$, \overrightarrow{CD} 平行于 \overrightarrow{BA} 。
- $\angle BCE > \angle BCD$, $\angle BCE > \angle DCE$ 。
- $\overrightarrow{DA} \times \overrightarrow{DE} > 0$, $\overrightarrow{AB} \times \overrightarrow{AD} > 0$, $\overrightarrow{BC} \times \overrightarrow{BA} > 0$, $\overrightarrow{CE} \times \overrightarrow{CB} > 0$, $\overrightarrow{ED} \times \overrightarrow{EC} > 0$ 。（其中 \times 运算符表示向量叉积）

RubyOnly 想知道，给定 n 个在二维平面中的点，从中选择 5 个不相同的点，能组成有多少个不相同的钻石。

两个钻石不相同当且仅当存在至少一个点的坐标不同。

Input

第一行包含 1 个整数 n ($1 \leq n \leq 300$)，表示点的个数。

接下来的 n 行，每行包含 2 个整数 x, y ($-10^8 \leq x, y \leq 10^8$)，用空格分隔，表示一个点。

保证不存在坐标相同的两个点。

Output

输出一行，包含一个整数，表示有多少个钻石。

Examples

standard input	standard output
5 0 0 3 3 3 -3 2 4 2 -4	1
9 0 0 0 1 0 -1 1 0 1 1 1 -1 -1 0 -1 1 -1 -1	4

Problem D. A xor B problem

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

Rubyonly有一个长度为 2^{32} 的数组 S ，以及 26 个变量 a, b, c, d, \dots, z ，均只能用于存储 U32^\dagger 类型的整数。

$^\dagger \text{U32}$ 为：一种必须满足大于等于 0 并且小于等于 $2^{32} - 1$ 的非负整数类型。

数组 S 的下标从 0 开始。我们用 $S[i]$ 表示数组 S 的下标为 $(i \bmod 2^{32})$ 的元素。

初始状态下，数组 S 所有元素均为 0，变量 a 存有初始值 A ($0 \leq A < 2^{16}$)，变量 b 存有初始值 B ($0 \leq B < 2^{16}$)，其余变量均为随机的 U32 整数。

Rubyonly希望把变量 c 所存储的值变成 A 和 B 的按位异或。为此，他打算进行若干次操作，每次操作均是以下 3 种类型之一。

- "1 U V W"：将变量 U 的值，赋值给 $S[\text{变量 } V + \text{变量 } W]$ 。（变量 U 、变量 V 以及变量 W 均是变量 a, b, c, d, \dots, z 中的一个，可以有重复）
- "2 U V W"：将 $S[\text{变量 } V + \text{变量 } W]$ 的值，赋值给变量 U 。（变量 U 、变量 V 以及变量 W 均是变量 a, b, c, d, \dots, z 中的一个，可以有重复）
- "3 U X"：将一个 U32 类型的整数 X ，赋值给变量 U 。（变量 U 是变量 a, b, c, d, \dots, z 中的一个，整数 X 是一个具体的 U32 整数）

请你帮Rubyonly找到一个操作序列，其中包含不超过 10^6 次操作，使得：无论 A 和 B 是多少，在按照顺序进行完操作序列中的操作后，变量 c 所存储的值，都能够变成 A 和 B 的按位异或。

可以证明，在本题的条件下，这样的操作序列一定存在，并且不唯一。

你只需要找到任意一个满足要求的操作序列并输出即可。

Input

本题没有输入。

Output

输出的第一行包含 1 个整数 q ($0 \leq q \leq 10^6$)，表示你的操作序列中一共使用了 q 次操作。

接下来的 q 行，按照顺序输出操作序列中的操作。

你可以参照下方样例输出中的例子。

Example

standard input	standard output
None	8 3 x 0 3 y 0 3 z 123 1 z y x 3 y 233 1 y x y 1 z y z 2 c y z

Note

注意：样例输出只是为了帮助更好的理解输出格式，实际不一定能通过此题。

在样例输出的解释如下。

第 1 步操作：将变量 x 赋值为 0 。

第 2 步操作：将变量 y 赋值为 0 。

第 3 步操作：将变量 z 赋值为 123 。

第 4 步操作：将 $S[y + x]$ 赋值为 z 。

第 5 步操作：将变量 y 赋值为 233 。

第 6 步操作：将 $S[x + y]$ 赋值为 y 。

第 7 步操作：将 $S[y + z]$ 赋值为 z 。

第 8 步操作：将 c 赋值为 $S[y + z]$ 。

Problem E. Rolling for the Destination

Input file: standard input
Output file: standard output
Time limit: 3 seconds
Memory limit: 512 megabytes

有一个无限长的数轴，Rubyonly初始位于数轴的 0 点处。

Rubyonly有一枚 4 面的骰子，每次投掷时，都会等可能的出现一个 1 到 4 之间的整数。也就是说，1, 2, 3, 4 这四个数各有 $\frac{1}{4}$ 的概率出现。

Rubyonly希望到达数轴上的整数点 n 。为此，Rubyonly将按照下列步骤进行操作。

1. 我们记Rubyonly当前所在的位置为 pos 。
2. 如果 $pos \geq n$ ，那么结束操作。
3. 否则，Rubyonly使用将骰子进行一次投掷。我们记投掷结果为 res 。
4. 之后，Rubyonly将会移动到 $(pos + res)$ 的位置。
5. 回到步骤 1，再次从步骤 1 开始，往后依次进行操作。

也就是说，在Rubyonly的所在位置大于等于 n 之前，Rubyonly将不断投掷骰子，并根据投掷结果向前行走。

请你帮Rubyonly计算一下，在他结束操作之后，他刚好位于整数点 n 的概率是多少。

请计算并输出这个概率对 998244353 取模后的结果。

更具体的讲，可以证明，这个概率一定能够被表示为一个分数 $\frac{p}{q}$ (p 和 q 都是正整数且互质)。你需要输出一个整数 x ，满足 $0 \leq x < 998244353$ 且 $x \cdot q \equiv p \pmod{998244353}$ 。

例如， $1/4$ 对 998244353 取模后的结果是 748683265。

Input

本题的输入包含多组测试数据。

输入的第一行包含 1 个整数 T ($1 \leq T \leq 1000$)，表示一共有 T 组测试数据。

对于每组测试数据：输入仅一行，包含 1 个整数 n ($0 \leq n \leq 10^{50000000}$)，表示Rubyonly的目标点。

保证测试数据的 n 的位数之和不超过 $5 \times 10^7 + 1$ 。

Output

对于每组测试数据，输出一行，包含一个整数 x ，表示所求的概率对 998244353 取模后的结果。

Example

[illegible]

Note

在第 1 组测试数据中, $n = 0$, 此时概率为 1。

在第 2 组测试数据中, $n = 1$, 此时概率为 $1/4$ 。

在第 3 组测试数据中, $n = 2$, 此时概率为 $5/16$ 。

在第 4 组测试数据中, $n = 3$, 此时概率为 $25/64$ 。

Problem F. 静流的路径

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

静流是一名可爱的女孩子，住在风祭市的她仰慕和喜欢着本题出题人。



Nakatsu Shizuru

静流有一个可以被分解质因数的正整数 N 。形式化的，我们有 $N = p_1^m \times p_2^m \times \cdots \times p_k^m$ ，其中 p_i ($1 \leq i \leq k$) 均为质数， m 为常数。

静流将正整数 y 分解质因数后的指数和记作 $f(y)$ 。形式化的，如果 $y = p_1^{b_1} \times p_2^{b_2} \times \cdots \times p_k^{b_k}$ ，那么我们有 $f(y) = b_1 + b_2 + \cdots + b_k$ 。特别的，我们规定 $f(1) = 0$ 。

一个正整数 y 被叫做**罚抄数**当且仅当 $f(y) = T$ 且 $y \mid N$ ，其中 T 为给定参数。

静流想要构建若干条从 1 到 N 的路径 $[x_1, x_2, x_3, \cdots, x_q]$ （其中 $2 \leq q \leq N$ 且 $1 = x_1 < x_2 < \cdots < x_q = N$ ）。

她只会考虑有这样性质的路径：对于所有满足 $1 \leq i \leq q-1$ 的 i ，都有 $x_i \mid x_{i+1}$ 成立。这里， $a \mid b$ 表示 a 整除 b 。不同的路径可以有部分数字重叠。

静流想要最小化构建的路径的数量，使得对于任意一个**罚抄数** c ， c 在某条路径中出现过。

请你帮静流计算一下，她至少需要构建多少条路径，才能满足这一要求。

由于答案可能很大，你只需要告诉静流这个数量模 998244353 的结果。

Input

输入的第一行包含 3 个整数 m, k, T ($1 \leq m, k \leq 10^3, 1 \leq T \leq m \times k$)，表示指数、质因数个数和参数。

输入第二行包含 k 个互不相同的质数 p_1, p_2, \dots, p_k ($2 \leq p_i \leq 10^6 + 3$)，表示 N 的所有质因数。

Output

输出一行，包含一个整数，表示所求的路径数模 998244353 的结果。

Example

standard input	standard output
2 3 3 2 3 7	7

Note

在样例一中，静流可以构建如下 7 条路径：

- [1, 2, 6, 18, 36, 1764]
- [1, 12, 84, 252, 1764]
- [1, 21, 63, 1764]
- [1, 7, 21, 147, 294, 882, 1764]
- [1, 3, 42, 294, 1764]
- [1, 14, 98, 294, 588, 1764]
- [1, 2, 14, 28, 84, 252, 1764]

Problem G. Candidate Master of Both (VI)

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

Rubyonly是一位同时精通位运算和数论的究极高手。今天，他给你出了这样一道题目。
给定一个长度为 n 的整数序列 a ，下标从 1 开始，元素分别记作 $a_1, a_2, a_3, \dots, a_n$ 。
我们定义 $f(l, r, k)$ 为：

$$f(l, r, k) = \begin{cases} \bigoplus_{i=l}^r a_i & , \text{ if } \gcd(l, r) = k \\ 0 & , \text{ if } \gcd(l, r) \neq k \end{cases}$$

其中， $\gcd(x, y)$ 表示整数 x 和 y 的最大公约数， \oplus 符号表示按位异或。
现在，你需要对于满足 $1 \leq k \leq n$ 的每个整数 k ，计算以下表达式的值：

$$\sum_{1 \leq l \leq r \leq n} f(l, r, k)$$

Input

输入的第一行包含 1 个整数 n ($1 \leq n \leq 10^5$)，表示序列的长度。
输入的第二行包含 n 个整数 $a_1, a_2, a_3, \dots, a_n$ ($0 \leq a_i \leq 10^5$)，用空格分隔，表示序列中的元素。

Output

输出一行，包含 n 个整数，用空格分隔，分别表示 $k = 1, 2, 3, \dots, n$ 时的答案。

Examples

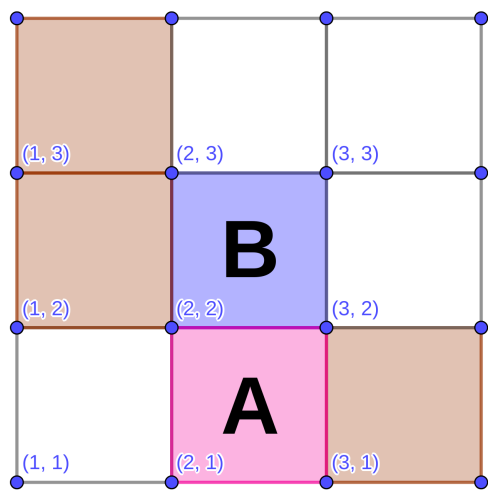
standard input	standard output
5 1 2 3 4 5	20 7 3 4 5
11 14 87 36 14 62 44 82 70 86 38 47	2821 787 324 22 180 44 82 70 86 38 47

Problem H. Duel on the Chessboard

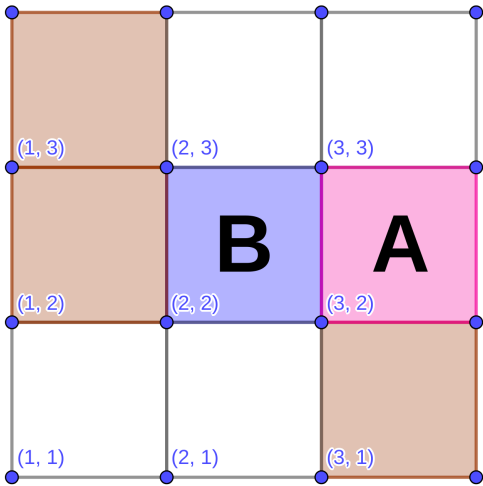
Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 256 megabytes

Frost_Ice与Febleaf两人正在下棋。
棋盘一共有 n 行 m 列。我们规定第 x 行与第 y 列所交的格子的坐标为 (x, y) 。
我们定义，如果两个位置 $A(x_1, y_1)$ 与 $B(x_2, y_2)$ 满足 $|x_1 - x_2| + |y_1 - y_2| = 1$ ，则我们称它们是相邻的。
棋盘上有两枚棋子 A 和 B ，它们初始是相邻的。
除此之外，棋盘上还分布有一些障碍物。障碍物上不能摆放棋子。
我们定义棋子的状态，为 A 的坐标 (x_A, y_A) 与 B 的坐标 (x_B, y_B) 构成的有序四元组 $[x_A, y_A, x_B, y_B]$ 。
Frost_Ice与Febleaf轮流进行操作，Frost_Ice先手。
每次操作，都需要选择棋子 A 和 B 中的一个，并将它围绕另一个棋子顺时针或逆时针旋转 90° 。
也就是说，假设选择的棋子的坐标为 (x_1, y_1) ，另一个棋子的坐标为 (x_2, y_2) ，则该次操作可以将其移动到 $(-y_1 + y_2 + x_2, x_1 - x_2 + y_2)$ 或者 $(y_1 - y_2 + x_2, -x_1 + x_2 + y_2)$ 。
同时，操作还需要满足以下条件的限制：

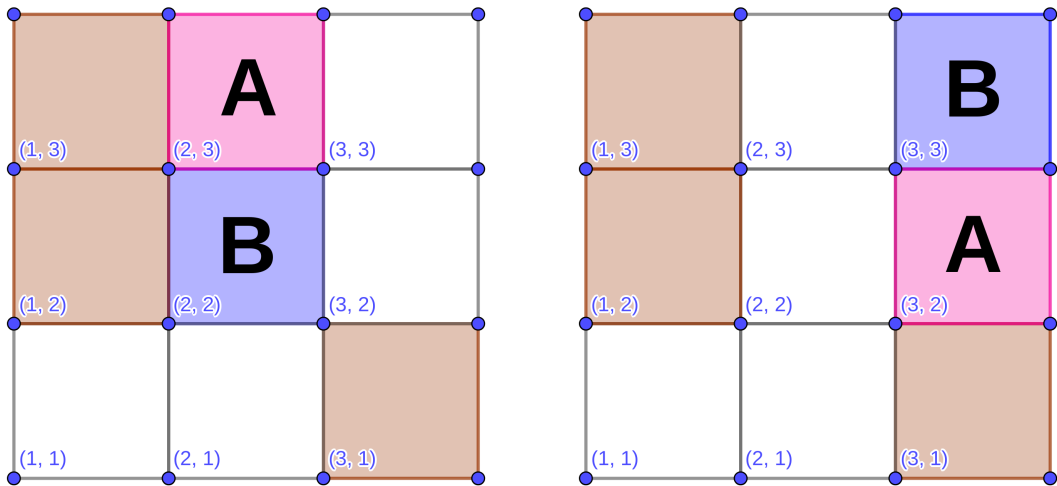
- 1. 棋子不能移动到棋盘外。
 - 2. 棋子不能移动到有障碍物的位置。
 - 3. 不能走重复的状态。也就是说，如果操作完成后的棋子的状态，在之前某个时刻已经出现过了，那么该操作就是不合法的。
- 举个例子，假设初始状态如下图所示，棋子 A 位于 $(2, 1)$ ，棋子 B 位于 $(2, 2)$ ， $(1, 2), (1, 3), (3, 1)$ 的位置有障碍物。



第一步，轮到Frost_Ice操作时，他可以将棋子 A 围绕 B 逆时针旋转 90° ，使其移动到 $(3, 2)$ 坐标。如下图所示。



那么第二步，轮到Febleaf操作时，他就只能将棋子 A 移动到 $(2,3)$ ，或者将棋子 B 移动到 $(3,3)$ 。如下图所示。他不能将棋子 A 移动回 $(2,1)$ ，因为该状态此前已经出现过了。



如果轮到某个人进行操作时，他无法进行操作，那么他就输了，另一方将获得胜利。
假设双方都采取最优的策略，请你计算一下最后谁会获得胜利。

Input

输入的第一行包含 2 个整数 n 和 m ($2 \leq n, m \leq 5000$, $4 \leq n \cdot m \leq 10^4$)，用空格分隔，表示棋盘的大小。
接下来 n 行，每行包含 m 个字符。
第 $(i + 1)$ 行的第 j 个字符表示棋盘 (i, j) 坐标位置的初始状态。
"A" 代表该位置为棋子 A 的初始位置。
"B" 代表该位置为棋子 B 的初始位置。
"#" 代表该位置有障碍物。
"." 代表该位置什么都没有。
保证初始状态下 A 和 B 一定相邻。

Output

输出一行，包含一个字符串，表示最终谁会获得胜利。

如果Frost_Ice会获得胜利，请输出 **Frost_Ice** 。

不喜欢Frost_Ice的同学也可以输出任意一个包含且只包含大写字母、小写字母、下划线的长度不超过 10 的字符串作为替代。例如 **Aa_** 。

否则，如果Febleaf会获得胜利，请输出 **Febleaf** 。

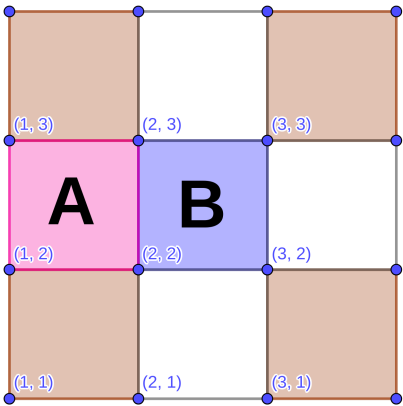
同样，不喜欢Febleaf的同学也可以输出任意一个包含且只包含大写字母、小写字母的长度不超过 10 的字符串作为替代。例如 **Aa** 。

Examples

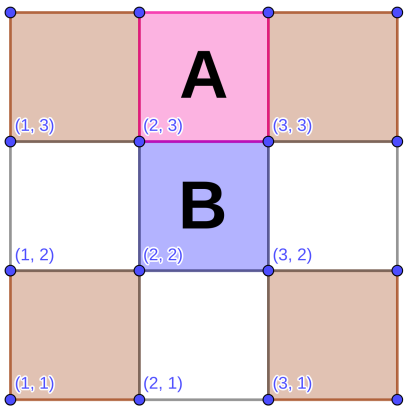
standard input	standard output
3 3 #.# AB. #.#	Frost_Ice
3 3 #.# AB. ###	Febleaf
4 5 .#.#.#.B. .#.A.	Febleaf
4 5# ####. A.#.. B...#	Frost_Ice

Note

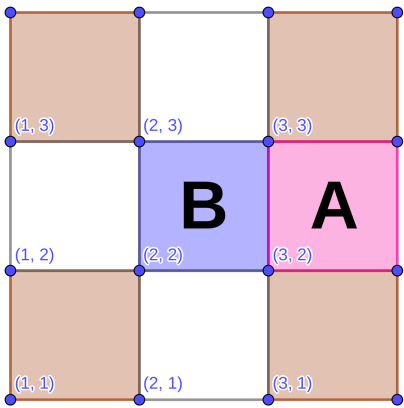
样例一中，初始状态如下。



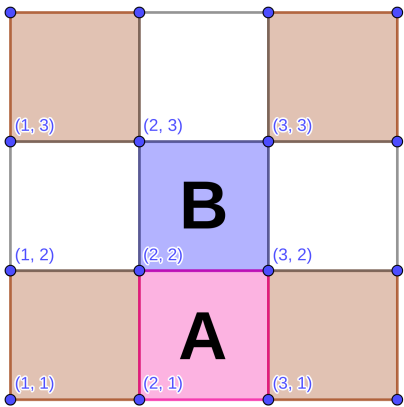
Frost_Ice可以先将棋子 A 移动到 (2,3) 。



随后，由于状态不能重复的限制，Febleaf此时只能将棋子 A 移动到 $(3,2)$ 。



同理，之后的一步，Frost_Ice只能继续将棋子 A 移动到 $(2,1)$ 。



最后，由于状态不能重复的限制，此时Febleaf无法进行操作，因此Frost_Ice获得胜利。

Problem I. 黄金树

Input file: **standard input**
Output file: **standard output**
Time limit: 1.5 seconds
Memory limit: 512 megabytes

故事发生在交界地。

黄金律法维持着交界地的秩序。"黑刀之夜"却造成了黄金律法的崩溃。

作为一名褪色者，你需要去集齐所有大卢恩，重建新的律法。



黄金树，以及黄金树笼罩之下的交界地

但是在此之前，你打算先去升级一下你的血瓶，以提高你的回血量。

升级血瓶，需要用到黄金树露滴。而黄金树露滴，产自黄金树。

黄金树是一棵由 n 个节点组成的有根树，节点分别编号为 $1 \sim n$ 。其中，黄金树之根的编号为 1。

黄金树的每个节点，每天都能产出一定数量的黄金树露滴。

我们记节点 i ($1 \leq i \leq n$) 在第 d 天 ($d \geq 0$) 的产出效率为 $P[i][d]$ 。

然而，随着黄金律法的衰弱，黄金树产出黄金树露滴的能力也在逐渐减弱。

在第 0 天，所有节点的产出效率都为正整数。之后，每过 1 天，都会有部分节点的产出效率发生减小。

具体来讲，在第 d 天，如果某个节点 i 满足以下两个条件之一：

- 节点 i 是黄金树之根。
- 节点 i 不是黄金树之根，且该节点的产出效率大于其父亲节点的产出效率。

那么，在第 $d+1$ 天，其产出效率就会减 1（最多减少到 0）。否则，其产出效率将保持不变。

形式化的，我们用 $fa[i]$ 表示节点 i 的父亲节点，则有：

$$P[i][d+1] = \begin{cases} \max(0, P[i][d] - 1) & , \text{ if } i = 1 \text{ or } P[i][d] > P[fa[i]][d] \\ P[i][d] & , \text{ else} \end{cases}$$

你非常关心你的血瓶的升级，因此你想知道，自己分别还能从每个节点那里收集到多少个黄金树露滴。

因此，请你对于每个节点 i ，计算 $\sum_{d=0}^{+\infty} P[i][d]$ 的值。

可以证明，其必定是一个有限的非负整数。

Input

本题的输入包含多组测试数据。

输入的第一行包含 1 个整数 T ($1 \leq T \leq 10^4$)，表示一共有 T 组测试数据。

对于每组测试数据：

输入的第一行包含 1 个整数 n ($2 \leq n \leq 10^6$)，表示黄金树的节点个数。

输入的第二行包含 n 个整数，用空格分隔，第 i 个整数 $P[i][0]$ ($1 \leq P[i][0] \leq 10^9$) 表示节点 i 在第 0 天的产出效率。

输入的第三行包含 $n-1$ 个整数，用空格分隔，第 i 个整数 $fa[i+1]$ ($1 \leq fa[i+1] \leq n$) 表示节点 $i+1$ 的父亲节点编号。

输入数据保证 n 个节点一定构成一棵有根树。

保证测试数据的 n 的总和不超过 10^6 。

Output

对于每组测试数据：

输出一行，包含 n 个整数，用空格分隔，第 i 个整数表示节点 i 的答案。

Example

standard input	standard output
6	3 3 5 4 15
5	36 30 35 43 40 48
2 1 2 1 5	28 15 9 13 11
1 1 2 2	28 33 40 51
6	190 496 377 228 90 351 1176 300 300 364
8 4 5 7 7 6	500000000500000000 1000000001
1 1 3 2 4	
5	
7 2 1 3 1	
1 2 3 4	
4	
7 6 7 9	
1 2 3	
10	
19 31 26 8 5 26 48 24 24 8	
1 6 2 1 8 8 1 5 7	
2	
1000000000 1	
1	

Note

在样例的第一组测试数据中：

节点	每一天的产出效率	总和
1	[2, 1, 0, 0, 0, 0, ...]	3
2	[1, 1, 0, 0, 0, 0, ...]	3
3	[2, 2, 1, 0, 0, 0, ...]	5
4	[1, 1, 1, 0, 0, 0, ...]	4
5	[5, 4, 3, 2, 1, 0, ...]	15

Problem J. 简单的指数运算

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

给定一个正整数 N 和一个质数 P 。
请你计算以下表达式的值：

$$\prod_{i=1}^N \prod_{j=1}^N (i \cdot j)^{d(i \cdot j)} \pmod{P}$$

其中， $d(x)$ 为 x 的正因数的数目。也就是说， $d(x) = \sigma_0(x) = \sum_{d|x} 1$ 。

Input

仅一行，包含两个正整数 N 和 P ($1 \leq N \leq 10^6, 10^7 + 19 \leq P \leq 10^9 + 7$)，用空格分隔，含义如题面所述。
输入数据保证 P 是质数。

Output

仅一行，包含一个整数，为上述表达式的值。

Examples

standard input	standard output
2 998244353	1024
114514 1000000007	925016843