

2025-2026 学年大连理工大学开发区校区 程序设计竞赛新生赛

辽宁，大连

2026 年 1 月 10 日



题号	题目名称
A	集合栈计算机 II
B	机械猫的流浪
C	跟我的漫游者说去吧
D	无人能证的猜想
E	Excel 的列编号
F	送熏肉
G	远交近攻
H	Silly Tree
I	奶龙塔
J	小猫钓鱼
K	用户名长度应在 5 位到 12 位之间

请勿在比赛正式开始前打开题面！

Problem A. 集合栈计算机 II

Input file: standard input
 Output file: standard output
 Time limit: 1 second
 Memory limit: 1024 megabytes

鉴于集合作为数学基础的重要性，一群理论家正着手构建一台基于集合而非数字的超级计算机。他们搞出的第一台原型机取名叫 Set Stack Alpha。理论家们希望对这台计算机进行一些测试，你需要根据他们的需求，给出一个测试用的指令序列。请注意，本题中的集合都是空集或集合的集合。

我们用大括号表示集合，比如说 $\{\}$ 是一个空集， $\{\{\}\}$ 是有一个空集作为元素的集合， $\{\{\}, \{\{\}\}\}$ 是一个包含前两者作为元素的集合。定义函数 $f(S)$ ，其中 S 是一个集合， $f(S)$ 的值是 S 写成上述格式后，括号对的数量。比如说对上述三个集合，有： $f(\{\}) = 1$ ， $f(\{\{\}\}) = 2$ ， $f(\{\{\}, \{\{\}\}\}) = 4$ 。

对于函数 f ，形式化的表述是

$$f(S) = \begin{cases} 1, & S = \{\} \\ 1 + \sum_{A \in S} f(A), & S \neq \{\} \end{cases}$$

计算机有一个栈，栈中的元素都是集合，栈初始为空。计算机支持如下五种操作，理论家们会给你一个正整数 x ，希望你能够给出一个不超过 250 条指令的指令序列，使得最后栈中仅剩一个元素 T ，且 $f(T) = x$ 。

- PUSH 指令会在栈顶推入一个空集 $\{\}$ 。
- DUP 指令会复制栈顶元素（等价于弹出栈顶元素，然后将其推入栈两次）。栈为空时是非法指令。
- UNION 指令会弹出栈顶元素两次，然后将两个弹出的集合的并集推入栈中。栈中元素少于两个时是非法指令。
- INTERSECT 指令会弹出栈顶元素两次，然后将两个弹出的集合的交集推入栈中。栈中元素少于两个时是非法指令。
- ADD 指令会弹出栈顶元素两次，将第一个弹出的集合加入第二个弹出的集合，然后将结果集合推入栈中。栈中元素少于两个时是非法指令。

为了详细说明，我们假设栈顶元素是：

$$A = \{\{\}, \{\{\}\}\}$$

栈中下一个元素是：

$$B = \{\{\}, \{\{\{\}\}\}\}$$

- 若执行 UNION 指令会从栈中弹出 A, B ，产生一个集合 $\{\{\}, \{\{\}\}, \{\{\{\}\}\}\}$ 入栈。

- 若执行 INTERSECT 指令会从栈中弹出 A, B ，产生一个集合 $\{\{\}\}$ 入栈。
- 若执行 ADD 指令会从栈中弹出 A, B ，产生一个集合 $\{\{\}, \{\{\}\}, \{\{\}, \{\{\}\}\}$ 入栈。

Input

仅有一行包含一个整数 x ($1 \leq x \leq 2^{63} - 1 = 9223372036854775807$)，表示理论家们希望最终得到的 $f(T)$ 。

Output

第一行输出一个整数 m ($1 \leq m \leq 250$) 表示操作序列的指令数量。

接下来 m 行，每行输出一个字符串代表指令，这个字符串必须是 PUSH, DUP, UNION, INTERSECT, ADD 其中之一。指令字符串的字母必须全部大写。

你的答案将被判定为正确，当且仅当指令序列合法，且按照输出的指令序列操作，最终栈中只剩下一个元素 T 且 $f(T) = x$ 。

Example

standard input	standard output
4	11 PUSH DUP PUSH DUP ADD ADD ADD DUP INTERSECT DUP UNION

Note

样例中，经过前四次操作，栈中有四个空集。

接下来三次操作，将栈中集合全部合并为 $\{\{\{\{\}\}\}\}$ 。

接下来两次操作，先复制再求交集，操作抵消，仅作输出格式展示。

接下来两次操作，先复制再求并集，操作抵消，仅作输出格式展示。

最终得到的集合是 $T = \{\{\{\{\}\}\}\}$ ， $f(T) = 4$ 。

Problem B. 机械猫的流浪

Input file: standard input
 Output file: standard output
 Time limit: 1 second
 Memory limit: 1024 megabytes

一只可怜的机械小猫正在野外流浪。小猫距离愿意收留它的营地还有很长的距离，可它的能量已经不够用了。幸运的是，小猫在沿途发现了一些定时开放的能量补给站。

具体而言：小猫与营地之间的路程可以视为一条长度为 $n + 1$ 的数轴，第 0 时刻，小猫在 1 位置，营地在 $n + 1$ 位置。小猫在到达营地前，每单位时间都需要消耗一格能量，维持自身机械的运作；如果它从当前位置走到下一个位置，需要一个单位的时间，并额外消耗一格能量。如果小猫在某时刻的能量小于等于零，它的机械会永久损坏，无法继续行动。

从 1 到 n 的每个位置上都有一座能量补给站，第 i 座能量补给站在 l_i 到 r_i 时间内（包括 l_i, r_i 两个时刻）开放，可以为小猫提供 a_i 格能量，并且每座补给站的能量只能提供一次。

小猫渴望着营地，因此它不会往回走，只能选择前进或者暂时停下。在机械不损坏的情况下，它需要多长时间到达营地？

Input

第一行两个正整数 n, E ($1 \leq n \leq 5 \times 10^3, 1 \leq E \leq 10^5$)，代表小猫到营地的路程长度与初始时剩余的能量格数。

第 2 至 $n + 1$ 行，第 $i + 1$ 行三个整数 a_i, l_i, r_i ($1 \leq a_i \leq 10^5, 1 \leq l_i \leq r_i \leq 10^4$)，代表第 i 座补给站在 $[l_i, r_i]$ 时间内提供 a_i 格能量。

Output

输出一个整数 t ，代表在机械不损坏的情况下，小猫到达营地的最短时间。如果无法到达，输出 -1 。

特别地，一旦小猫的能量小于等于 0，则它的机械立刻损坏，即使它在该时刻刚好能获得补给或者到达营地，仍然视为无法到达。

Example

standard input	standard output
4 3 8 3 4 5 1 2 8 5 6 8 5 6	7

Problem C. 跟我的漫游者说去吧

Input file: standard input
 Output file: standard output
 Time limit: 1 second
 Memory limit: 1024 megabytes

很多游戏里面都有暴击的设定，当你打出一个暴击，突然的音效与高额的对敌伤害会令你激动不已。然而，运气问题总是令人十分头疼。如果设定暴击率为 50%，但连续四五下打不出暴击，不仅令人沮丧，还可能会改写一场战局的结果。为了避免连续太多次打不出暴击的情况，我们可以采用动态暴击率的解决方案，给暴击一个保底。

下面给出一种简单的动态暴击率实现方案。定义暴击系数为 c ($0 < c \leq 1$)，定义 i 为本次攻击的最近暴击距离（意思是，若之前有过暴击，则本次攻击是上次暴击后的第 i 次攻击；否则本次攻击是游戏内第 i 次攻击），则本次攻击有 $\min(1, ic)$ 的概率触发暴击。

例如，令 $c = 0.6$ 。对于某次攻击，若其为第一次攻击或者上一次刚刚暴击过，则该次攻击有 $\min(1, c) = 0.6$ 的概率暴击，否则有 $\min(1, 2c) = 1$ 的概率暴击，由此我们就解决了暴击的无保底问题。

令 $f(n)$ 代表从游戏开始连续进行的 n 次攻击中暴击数量的期望值。于是我们可以定义 $p = \lim_{n \rightarrow \infty} \frac{f(n)}{n}$ 为期望暴击率。

游戏策划已经搞出了期望暴击率的具体数值，只需要你算出对应的暴击系数 c 就可以啦！

Input

每个测试点包含多个测试用例。第一行一个整数 t ($1 \leq t \leq 100$)，代表测试用例的数量。

接下来 t 行，每行一个实数 p ($0.01 \leq p \leq 1$)，代表需要达成的期望暴击率。保证 p 是一个小数，且小数点后有 6 位数字。

Output

对于每个测试用例，输出一行一个实数 c ($0 < c \leq 1$)，代表达成期望暴击率所需的暴击系数。

允许的绝对误差不应该超过 10^{-8} 。形式化的表述是，设你的答案为 X ，标准答案是 S ，只有当你的答案满足 $|X - S| \leq 10^{-8}$ 时才会被判定为正确。

Example

standard input	standard output
4	0.000156041691677
0.010000	0.019170447447532
0.114514	0.7500000000000000
0.800000	1.0000000000000000
1.000000	

Problem D. 无人能证的猜想

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 1024 megabytes

任意选择一个正整数 x ，不断对其进行如下操作：

- 若 x 为奇数，将 x 乘 3 再加 1
- 若 x 为偶数，则将 x 除以 2

我们猜想，对于任意一个初始正整数 x ，它最终都会落入 $1 \rightarrow 4 \rightarrow 2 \rightarrow 1$ 的循环当中。

例如，以 6 作为初始值，发现 $6 \rightarrow 3 \rightarrow 10 \rightarrow 5 \rightarrow 16 \rightarrow 8 \rightarrow 4 \rightarrow 2 \rightarrow 1$ ，确实进入了这个循环。

上述猜想至今未被证明或证伪。不过本题只希望你完成这样一个任务：对于给定的初始 x 值，计算出使 $x = 1$ 的最小操作次数。

在本题涉及的数据范围内，上述猜想为真。

Input

输入仅有一行一个正整数 x ($1 \leq x \leq 2^{31} - 1$)。

输入数据保证计算过程中 x 不会超过 $2^{31} - 1$ 。保证答案存在且小于 1000。

Output

输出一个非负整数 t 代表对于给定的初始 x 值，使 $x = 1$ 的最小操作次数。

Examples

standard input	standard output
1	0
6	8
27	111

Note

出题人想要提醒你，如果你不知道如何合理地使用与加速 C++ 的输入和输出，就尽可能不要使用它们。尤其是不要使用 `std::endl`。当然，本题数据量很小，不必担心 IO 导致的超时。

Problem E. Excel 的列编号

Input file: standard input
 Output file: standard output
 Time limit: 1 second
 Memory limit: 1024 megabytes

X	Y	Z	AA	AB	AC
24	25	26	27	28	29
ZX	ZY	ZZ	AAA	AAB	AAC
700	701	702	703	704	705

你有没有观察过 Excel（或 WPS 表格）的列编号？为了防止与行混淆，Excel 使用大写字母字符串对列进行编号。

- 第 1 列到第 26 列，使用一个字母，编号为从 A 到 Z
- 接下来 26^2 列，使用两个字母，编号为 AA, AB, ..., AZ, BA, BB, ..., BZ, CA, ..., ZZ
- 接下来 26^3 列，使用三个字母，编号为从 AAA 到 ZZZ，方法同上
-

你的任务是，对于给定的数字 a ，输出第 a 列对应的列编号。

Input

每个测试点包含多个测试用例。第一行一个正整数 t ($1 \leq t \leq 10^5$) 代表测试用例的数量。

接下来 t 行，每行一个正整数 a_i ($1 \leq a_i \leq 10^{18}$)，代表对第 a_i 列的编号进行询问。

Output

对于每个测试用例，输出一行一个字符串，代表对应的列编号。

Example

standard input	standard output
5	E
5	AA
27	AZ
52	BA
53	AAA
703	

Problem F. 送熏肉

Input file: standard input
Output file: standard output
Time limit: 2 seconds
Memory limit: 1024 megabytes

《空洞骑士：丝之歌》作为《空洞骑士》的续作，收获了 CPC 组的“一致好评”。其中的“送熏肉”任务，更是让玩家们的“赞叹连连”。

伴随着悠扬的歌声“三百六十五里路哟，越过春夏秋冬……”，小 X 的第 100 次“送熏肉”任务失败了。恼羞成怒的他决定利用游戏漏洞卡关——跳过一段连续的关卡，但这个漏洞仅能使用一次，否则会被 Team Cherry 发现并修复。

小 X 精心挑选了一段连续的高难度关卡，美美地跳过了它们。可他又傻眼了：下一个关卡的难度突然变高！伴随着悠扬的歌声“三百六十五里路哟，越过春夏秋冬……”，小 X 的第 101 次“送熏肉”任务失败了。

经过研究，小 X 发现“送熏肉”任务路径上的某个关卡难度会不定期变化，这意味着他之前选好的最优跳过关卡区间，可能会随着关卡难度的变化而不再最优！

他很快构思出了一套新算法，但因要去吃饭，便将编写代码实现该算法的任务交给了聪明的你。



给定一个长度为 n 的数组 a_1, a_2, \dots, a_n ，其中 a_i 表示“送熏肉”任务中第 i 个关卡的难度。接下来会依次经历 m 个时间段，每个时间段内恰好有一个关卡的难度被修改（修改后的值会保留，影响后续所有操作）。

对于初始的数组，以及每一次修改后的数组，请你选择一个长度恰好为 k 的连续子区间，跳过该子区间对应的 k 个连续关卡，使得剩余所有关卡的难度之和最小。请输出这个最小的和。

长度为 k 的连续子区间的定义为：若子区间的起始位置为 l ，则其结束位置为 $r = l + k - 1$ ，其中 $1 \leq l \leq n - k + 1$ ，该子区间包含的元素为 a_l, a_{l+1}, \dots, a_r 。

本题数据量很大，请选择较高效率的 IO 方式。

Input

第一行输入三个整数 n, m, k ($1 \leq n, m \leq 5 \times 10^5, 1 \leq k \leq n$), 分别表示关卡的总数、难度修改的次数、可跳过的连续关卡的长度。

第二行输入 n 个整数 a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$), 表示初始状态下每个关卡的难度。

接下来的 m 行, 每行输入两个整数 x, y ($1 \leq x \leq n, 1 \leq y \leq 10^9$), 表示将第 x 个关卡的难度修改为 y 。每次修改的结果会被保留, 且会影响后续所有查询的答案。

Output

输出共 $m + 1$ 行, 每行一个整数。其中第 i 行的整数对应: 经过前 $i - 1$ 次修改后, 选择跳过一段长度恰好为 k 的连续关卡, 剩余所有关卡难度之和的最小值。

Example

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

Problem G. 远交近攻

Input file: standard input
Output file: standard output
Time limit: 1.5 seconds
Memory limit: 1024 megabytes

有从左到右 n 个地块，编号为 $1 \sim n$ ，第 i 个地块上有 a_i 个单位的资源。起初，有 n 个部落，第 i 个部落生活在第 i 个地块上，并且占有其资源。随着时间的推移，不同部落之间开始交流，每个部落的首领都会遵循远交近攻的原则与其他部落相处。

具体而言，每个部落的领地是该部落当前占有的所有地块的集合（可能非连续）。

- 如果正在交流的两个部落之间领地不相邻（即：不存在地块 $x \in$ 部落 A 领地、地块 $y \in$ 部落 B 领地，使得 $|x - y| = 1$ ），他们会选择与对方结盟，合并变成一个部落，合并双方的领地与全部资源，并且由于合作额外获得 c 个单位的资源。
- 否则，他们之间会爆发战争，合并变成一个部落，合并双方的领地与全部资源，并且由于战争额外损失 d 个单位的资源。

已知按时间排序的所有交流事件，你需要随时知道任何一个部落的现有资源情况。

本题数据量很大，请选择较高效率的 IO 方式。

Input

第一行三个整数 n, c, d ($1 \leq n \leq 5 \times 10^5, 0 \leq c, d \leq 10^9$)，代表地块的数量，合作获得的资源量以及战争损耗的资源量。

第二行 n 个整数 a_1, a_2, \dots, a_n ($0 \leq a_i \leq 10^9$)， a_i 代表第 i 个地块原有的资源。

第三行一个正整数 q ($1 \leq q \leq 5 \times 10^5$)，代表操作次数。

第 4 至 $q+3$ 行，每行是以下两类操作之一：

- $1 \ x \ y$ 代表占有 x, y 两个地块的部落发生交流；特别地，如果 x, y 地块原本就属于同一个部落，则不会发生结盟或战争，本次操作无效。
- $2 \ x$ 代表询问占有 x 地块的部落的资源数量。

对所有操作，保证 $1 \leq x, y \leq n$ 。

Output

对于每个第二类操作，输出一行一个整数 w ，代表询问时该部落的现有资源数量。注意资源数量可能小于 0。

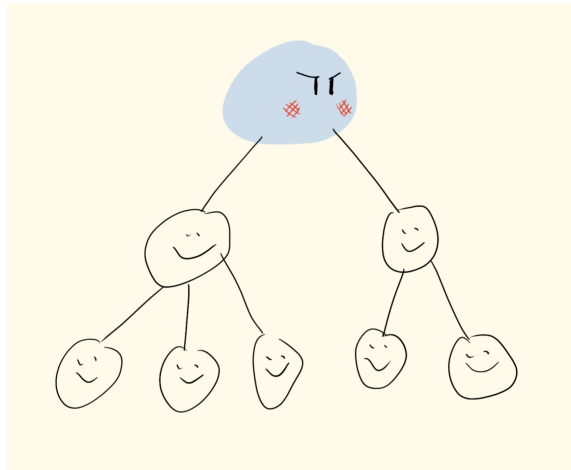
Example

standard input	standard output
5 2 3	9
1 2 3 4 5	10
4	
1 5 2	
2 2	
1 4 2	
2 5	

Problem H. Silly Tree

Input file: standard input
Output file: standard output
Time limit: 3 seconds
Memory limit: 1024 megabytes

某人在做题时，为了给队友讲解思路，画了一棵二叉树。然而，他没有注意到的是，他画出的某个节点有足足 3 个儿子。



我们定义符合如下所有条件的树是一棵 Silly Tree:

- 有一个根节点
- 每一个节点的儿子数量 c 都满足 $c \in A$
- $0 \in A$ 且 $A \subset \mathbb{N}$

与二叉树类似，每个节点的多个儿子是有序的。即，如果某两个儿子的子树不同，交换两个儿子后得到的树就与原来不同。

令 $k = |A|$ ，给定 n 和 A ，你要求出恰好有 n 个点的 Silly Tree 有多少种。由于答案可能很大，你需要对 998 244 353 取模。

Input

每个测试点包含多个测试用例。第一行一个整数 tc ($1 \leq tc \leq 3000$)，代表测试用例的数量。每个测试用例有两行输入:

- 第一行输入两个正整数 n, k ($1 \leq n \leq 3000, 1 \leq k \leq 10$)。
- 第二行输入 k 个互不相同的、单调递增的非负整数 a_1, a_2, \dots, a_k ($0 \leq a_i < n$) 代表集合 A ，保证 $a_1 = 0$ 。

保证所有测试用例的 n 之和不超过 3000。

Output

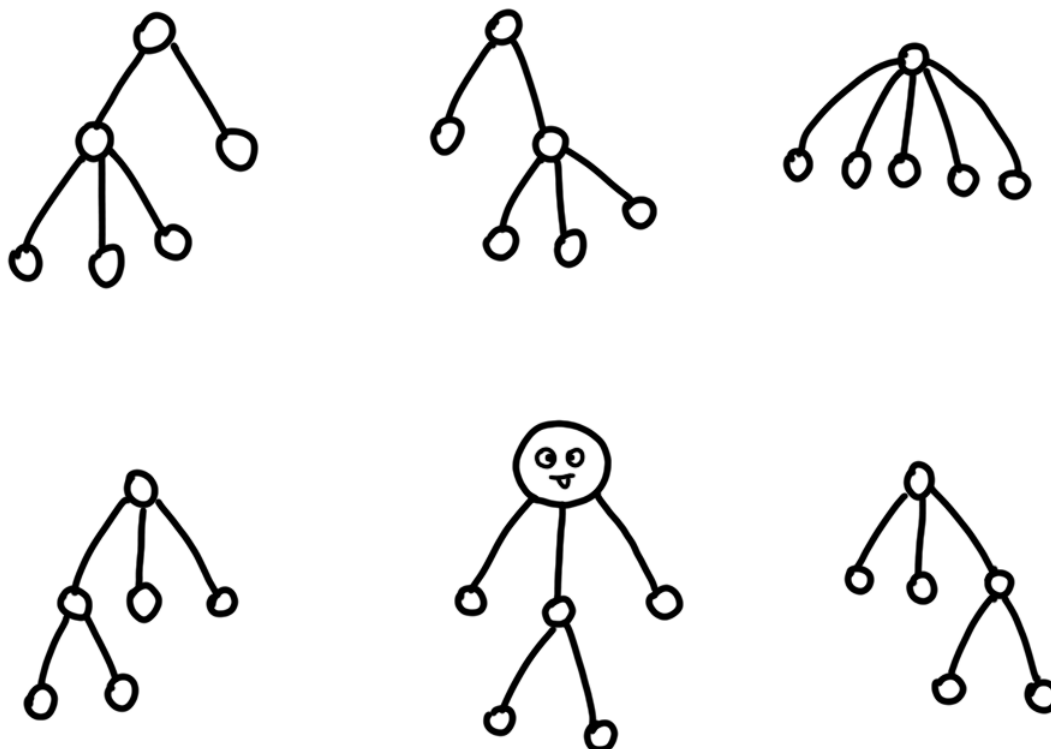
对每个测试用例输出一行一个整数，代表答案对 998 244 353 取模的结果。

Example

standard input	standard output
5	0
10 1	2
0	5
5 3	6
0 2 3	176899997
6 3	
0 2 3	
6 4	
0 2 3 5	
2026 8	
0 2 3 5 7 114 514 1001	

Note

对第四个测试用例的样例解释如下，共 6 种树。



Problem I. 奶龙塔

Input file: standard input
Output file: standard output
Time limit: 1 second
Memory limit: 1024 megabytes

A316 里有很多可爱的奶龙，有一天大家突发奇想，决定把所有奶龙堆在一起，形成一座壮观的奶龙塔。



搭建奶龙塔的过程中，聪明的 lingo12321 发现，每只奶龙只能堆在体积 **严格大于**自身体积的奶龙上，否则奶龙塔会因为重心不稳而倒塌。已知 A316 里的奶龙数量与每只奶龙的体积，lingo12321 想知道，他能不能将所有奶龙放进同一座奶龙塔里？

Input

第一行一个正整数 n ($1 \leq n \leq 2 \times 10^5$)，代表 A316 奶龙的数量。

第二行 n 个正整数 a_1, a_2, \dots, a_n ($1 \leq a_i \leq 10^9$)，代表每只奶龙的体积。

Output

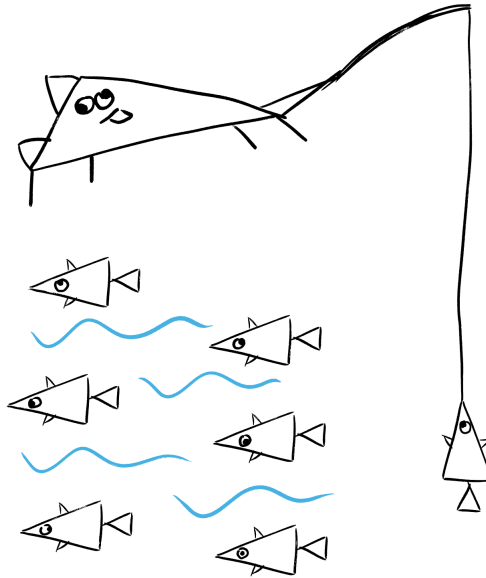
输出一行一个字符串，如果 lingo12321 可以将所有奶龙放进同一座奶龙塔，输出 “Yes”（没有引号），否则输出 “No”（没有引号）。

Examples

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

Problem J. 小猫钓鱼

Input file: standard input
 Output file: standard output
 Time limit: 2 seconds
 Memory limit: 1024 megabytes



“小猫钓鱼”是一种轻松愉快的扑克牌小游戏，它版本众多，不过 InterRiver 不喜欢其中的任何一个。他决定搞个自己的版本，规则如下。

“小猫钓鱼”是一种适合单人或多人游玩的小游戏。

游戏使用普通扑克牌中除了大小王的全部 52 种普通牌，每种扑克牌出现的次数没有限制。每张牌都有两项信息：花色与点数。

- 花色有 4 种：方片，梅花，红桃和黑桃，分别用字母 D, C, H, P 表示。
- 点数有 13 种：2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K, A。为了方便，用字母 T 代表数字 10。

用“花色 + 点数”的方式，任何一种普通牌都可以表示为一个长度为 2 的字符串，如 D3 代表方片 3，HK 代表红桃 K，PT 代表黑桃 10，C4 代表梅花 4。

n 名玩家参与游戏，将他们编号为 $1 \sim n$ ，每名玩家有一个分数。游戏使用回合制，每回合有且仅有一名玩家进行操作，其中第 i 回合的操作者为玩家 $[(i - 1) \bmod n] + 1$ 。例如，若 $n = 3$ ，游戏开始后的操作者分别为 1, 2, 3, 1, 2, 3, 1, 2, 3, ... 直到游戏结束。

游戏有三个区域：牌堆、池塘、弃牌堆。池塘是一个从左到右的长条形区域，游戏主要在这里进行。

游戏开始前，牌堆中有 m 张牌，弃牌堆和池塘中都没有牌，所有玩家的分数均为零。

游戏开始后，在每一个回合中，下述事件依次发生：

- 下竿：该回合操作者从牌堆顶上拿起一张牌，放到池塘的末尾。“末尾”指当前池塘中现有牌的右侧（最右端），即新牌会被放在现有牌的最右边，形成从左到右的有序序列。

- 钓鱼：若池塘中存在两张点数相同的牌（花色可以相同或相异），则将这两张牌与它们之间的牌从池塘转移到弃牌堆。这个被转移的扑克牌序列被称为“鱼”。
- 计分：若该回合发生了“钓鱼”事件，则根据规则对鱼进行计分，将分数加到该回合操作者身上。

弃牌堆中的牌永远不会再被抽取或使用。游戏不断进行，直到牌堆为空，即游戏总共有 m 个回合。

鱼的计分规则为：数出序列中的扑克牌张数 l ，再数出序列中不同花色的数量 c ，通过以下表格算出分数。注意：本题末尾有可供复制的表格。

	$c = 1$	$c = 2$	$c = 3$	$c = 4$
$l = 2$	21	13	—	—
$l = 3$	34	16	19	—
$l = 4$	47	22	18	32
$l = 5$	60	30	20	28
$l = 6$	74	39	25	27
$l = 7$	89	49	31	29
$l = 8$	106	60	38	33
$l = 9$	123	72	47	39
$l = 10$	148	86	57	47
$l = 11$	166	102	70	56
$l = 12$	189	119	84	68
$l = 13$	216	141	102	83
$l = 14$	248	167	125	105

已知某局游戏的 n 和 m ，以及初始牌堆从顶到底的每一张牌的信息。InterRiver 想知道每次计分事件的详细信息，以及每名玩家最终获得的总分。

本题数据量很大，请选择较高效率的 IO 方式。

Input

第一行输入两个正整数 n, m ($1 \leq n \leq 24, 1 \leq m \leq 2 \times 10^6$)，代表参与游戏的人数与初始牌堆大小。

第二行 m 个长度为 2 的字符串，依次代表初始牌堆从顶到底的每一张牌。每两个字符串间用一个空格隔开。字符串含义见题目描述部分。

Output

首先，按照时间顺序输出每次计分事件的信息。对于每次计分事件，输出一行两个整数 t 和 s ，中间用一个空格隔开，分别代表该回合操作者编号与其获得的分数。

最后，输出一行 n 个整数 a_1, a_2, \dots, a_n ，每两数之间用一个空格隔开，其中 a_i 代表游戏结束后第 i 名玩家的总分。

Examples

standard input	standard output
3 6 HT HT CA HT P4 PA	2 21 3 18 0 21 18
5 7 CA CT P3 C7 C8 PK CJ	0 0 0 0 0

Note

去除首行和首列，以逗号作为间隔符的表格体如下：

```
21, 13, -, -,  
34, 16, 19, -,  
47, 22, 18, 32,  
60, 30, 20, 28,  
74, 39, 25, 27,  
89, 49, 31, 29,  
106, 60, 38, 33,  
123, 72, 47, 39,  
148, 86, 57, 47,  
166, 102, 70, 56,  
189, 119, 84, 68,  
216, 141, 102, 83,  
248, 167, 125, 105,
```

Problem K. 用户名长度应在 5 位到 12 位之间

Input file: standard input
 Output file: standard output
 Time limit: 3 seconds
 Memory limit: 1024 megabytes

international_river 正在注册某个做题网站，他喜欢自己的这个名字。如果网站允许用户名是中文，他就叫“国际河流”，否则就叫 international_river。

但是，他遇到了困难！该网站提示：用户名由小写字母及下划线组成，且长度应在 5 位到 12 位之间。这可不好，因为 international_river 足足有 19 个字符。

于是，他决定缩减自己的用户名。缩减方法如下：

令 s 为原来的用户名，给定正整数 l, r ，保证 $2 \leq l \leq r < |s|$ 。选择 s 的一个非空前缀 a 与 s 的一个非空后缀 b ，满足 a 的最后一个字符等于 b 的第一个字符，并且 a 与 b 的长度之和 len 满足 $len \in [l, r]$ ，则缩减后的用户名 $t = a + b$ 。

比如，在上述例子中， $s = \text{international_river}$ ， $l = 5$ ， $r = 12$ ，则可以选择 $a = \text{inter}$ ， $b = \text{river}$ ，则缩减后的用户名 $t = a + b = \text{interriver}$ 。

international_river 想问你，对于给定的一个只有小写字母和下划线的字符串 s 以及给定的 l, r ，有多少种本质不同的符合要求的字符串 t ？

Input

每个测试点包含多个测试用例。第一行一个整数 tc ($1 \leq tc \leq 5000$)，代表测试用例的数量。

对每个测试用例，第一行输入一个只包含小写字母与下划线的字符串 s ($3 \leq |s| \leq 2 \times 10^4$)，代表原用户名；第二行输入两个正整数 l, r ($2 \leq l \leq r < |s|$)，表示缩减后的用户名的长度限制。

保证所有测试用例的 $|s|$ 的和不超过 2×10^4 。

Output

对每个测试用例，输出一行一个整数，表示本质不同的符合要求的字符串 t 的种类数。

Example

standard input	standard output
4	6
international_river	0
5 12	2
love_acm	9
2 7	
oiiiaiioooooiaii	
6 6	

2 10	

Note

对于第一个测试用例，有以下 6 种不同的字符串 t ：

- `i + iver = iiver`
- `inte + er = inteer`
- `inter + r = interr`
- `inter + river = interriver`
- `in + nal_river = innal_river`
- `i + ional_river = iional_river`

对于第二个测试用例， s 中没有相同的字符。

对于第三个测试用例，虽然有 3 种方案产生字符串 t ，但是有两种方案产生的字符串相同，不重复计数。