



# 2022 年第四届河南省 CCPC 大学生程序设计竞赛 正式赛

2022 年 10 月 2 日



## 题目概况

	题目名	时间限制	空间限制
A	Mocha 上小班啦	1 s	512 MB
B	Hash	2 s	512 MB
C	Serval 的试卷答案	1 s	512 MB
D	Mocha 上中班啦	2 s	512 MB
E	Serval 的俳句	1 s	512 MB
F	集合之和	1 s	512 MB
G	Mocha 上大班啦	1 s	512 MB
H	旋转水管	1 s	512 MB
I	Oshwiciqwq 的电梯	1 s	512 MB
J	Mex Tree	2 s	256 MB
K	复合函数	1 s	512 MB
L	串串串串……	10 s	512 MB



## Problem A. Mocha 上小班啦

输入文件: standard input

输出文件: standard output

时间限制: 1 second

空间限制: 512 megabytes

旭丘幼儿园的小班开设了一门教授数论的课程。Mocha 在研究数论的过程中, 发现了一种奇妙的数——Mocha 数。Mocha 数是一个由几个互不相同的数字构成且不含前导零的正整数。

为了方便研究, Mocha 想知道是否存在一个包含  $n$  个数位的 Mocha 数, 如果存在, 其中最小的 Mocha 数是多少。

### 输入格式

一个整数  $n$  ( $1 \leq n \leq 20$ ), 代表询问的位数。

### 输出格式

如果存在包含  $n$  个数位的 Mocha 数, 输出最小的  $n$  位 Mocha 数, 否则输出  $-1$ 。

### 样例

standard input	standard output
1	1

### 提示

1 到 9 都是只包含 1 个数位的 Mocha 数, 其中 1 是最小的 Mocha 数。

## Problem B. Hash

输入文件: standard input  
输出文件: standard output  
时间限制: 2 seconds  
空间限制: 512 megabytes

Bellalabella 最近学习了 Hash 算法。简单来说, 对于一个字符串  $S = s_1s_2 \dots s_{|S|}$ , 定义其 Hash 函数为

$$f(S) = \left( \sum_{i=1}^{|S|} g(s_i) \times 31^{|S|-i} \right) \bmod 998\,244\,353$$

由于 Bellalabella 十分喜欢河南的各种美食以及河南的队友 (确信), 因此她将字符串  $S$  的字符限定在了 **a, e, h, n** 四个字母, 相对应地有  $g(\mathbf{a}) = 1, g(\mathbf{e}) = 2, g(\mathbf{h}) = 3, g(\mathbf{n}) = 4$ 。

现在 Bellalabella 有一个仅包含 **h, e, n, a** 四个字母的字符串  $T = t_1t_2 \dots t_{|T|}$  且字符串  $T$  是一个环, Bellalabella 想将其分成若干个子串。

形式化地说, Bellalabella 可以选择任意个数个分割点  $1 \leq a_1 < a_2 < \dots < a_m \leq |T|$ , 这  $m$  个分割点可以将  $T$  分割成  $m$  个子串  $T[a_i \bmod |T| + 1, a_{i+1} \bmod |T| + 1]$ , 其中  $T[L, R] = t_L t_{L+1} \dots t_R$ , 特别地当  $L > R$  时  $T[L, R] = t_L t_{L+1} \dots t_{|T|} t_1 t_2 \dots t_R$ 。

例如  $T = \mathbf{henan}$ , 分割点为 1, 2, 4, 则串  $T$  会被分割为 **e, na, nh**。

Bellalabella 将串  $T$  分成若干个子串后, 每个子串都有自己的 Hash 值。Bellalabella 想知道在不同分割方法下, 所有子串 Hash 值之和最大为多少。

### 输入格式

仅一行, 一个仅包含 **a, e, h, n** 四个字母的字符串  $T$  ( $1 \leq |T| \leq 2 \times 10^5$ )。

### 输出格式

输出仅一行, 包含一个整数, 表示字符串  $|T|$  在不同分割方法下, 所有子串 Hash 值之和的最大值。

### 样例

standard input	standard output
henan	3785504

### 提示

输入字符串  $T$  为 **henan**, 当分割点为 4 时被分割成 **nhena**, Hash 值之和为 3785504 取到最大值。

## Problem C. Serval 的试卷答案

输入文件: standard input  
输出文件: standard output  
时间限制: 1 second  
空间限制: 512 megabytes

Serval 是加帕里幼儿园的新生。

期中考试前一天晚上, Serval 在梦中见到了考试试卷和试卷答案。

Serval 梦中的考试试卷只有不定项选择题, 题号从 1 开始标号, 但是题目数量并不确定。一道不定项选择题只有 ABCD 四个选项, 其答案是正确选项按字典序排列得到的非空字符串, 这意味着一道不定项选择题至少有一个正确选项。例如 ABD 或者 C 可以是一道不定项选择题的答案, 而 DA 以及 BB 不是。一张试卷的答案是不定项选择题答案连接而成的字符串。

梦醒之后, Serval 连忙写下了记忆中残留的答案片段。Serval 可能会修改写下的答案字符串, 但他最好奇的是, 对于某一段答案字符串, 有多少份恰好包含  $k$  道不定项选择题的不同试卷, 其答案恰为这个字符串。两份试卷不同当且仅当某道题号相同的不定项选择题的答案不同。

具体来说, Serval 写下了长度为  $n$  的仅包含 ABCD 的字符串  $S$ 。Serval 给出了  $q$  次操作, 操作分为以下两种:

1. 给出  $l, r$ , 对于  $l \leq x \leq r$ , 将  $S_x$  变为下一个选项, 即 A 变为 B, B 变为 C, C 变为 D, D 变为 A。
2. 给出  $l, r, k$ , 对于  $S_l, S_{l+1}, \dots, S_r$  依次连接得到的答案字符串, 求出有多少份恰好包含  $k$  道不定项选择题的不同试卷, 其答案恰为这个字符串。

对于第二个操作, 由于答案可能很大, 你只需要求出答案对 998 244 353 取模的结果。

Serval 还是幼儿园的小朋友, 并不能解决这个问题。你能帮他解决他的小问题, 满足他的好奇心吗?

### 输入格式

第一行, 两个整数  $n, q$  ( $1 \leq n \leq 10^5, 1 \leq q \leq 10^5$ ), 表示 Serval 写下的答案字符串的长度, 以及 Serval 给出的操作次数。

第二行, 长度为  $n$  的字符串  $S$ , 表示 Serval 写下的答案字符串。

接下来  $q$  行, 每行第一个整数  $op$  ( $1 \leq op \leq 2$ ), 表示 Serval 给出第  $op$  种操作。当  $op = 1$  时, 之后包含两个整数  $l, r$  ( $1 \leq l \leq r \leq n$ ), 表示操作参数。当  $op = 2$  时, 之后包含三个整数  $l, r, k$  ( $1 \leq l \leq r \leq n, 1 \leq k \leq n$ ), 表示操作参数。

### 输出格式

对于第二个操作, 输出一行一个整数, 表示所求的答案对 998 244 353 取模的结果。

## 样例

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

## 提示

对于样例，我们给出如下说明：

1. 仅有以下一种恰好包含 2 道不定项选择题的试卷的答案为 BACD:
  - B ACD
2. 答案字符串变为 BBDDA。
3. 仅有以下一种恰好包含 4 道不定项选择题的试卷的答案为 BBDDA:
  - B BD D A
4. 答案字符串变为 BCADA。
5. 有以下两种恰好包含 4 道不定项选择题的试卷的答案为 BCADA:
  - B C AD A
  - BC A D A
6. 仅有以下一种恰好包含 5 道不定项选择题的试卷的答案为 BCADA:
  - B C A D A

## Problem D. Mocha 上中班啦

输入文件: standard input  
输出文件: standard output  
时间限制: 2 seconds  
空间限制: 512 megabytes

在旭丘幼儿园, 有一个为中班小朋友开设的实验室, 但是实验室总是有许多杂物, 白板前面的票箱就是其中之一。

票箱有一个投票的开口。一天, Mocha 突发奇想, 她想看看能不能将白板笔从开口丢进去, 可 Mocha 尝试了许多次总是被挡住。然而当 Mocha 尝试旋转白板笔时, 突然白板笔掉进去了! Mocha 发现当白板笔处于某些角度时就可以丢进票箱, 否则就会被挡住。

爱动脑的 Mocha 当然不会放过这个神奇的现象! 她将这件事情抽象成了下面的模型:

二维平面上有一个  $n$  个点的凸多边形, 一个位于凸多边形内部或边界上的旋转中心, 以及一对平行的直线。Mocha 想请你计算一下, 如果她每秒将凸多边形绕旋转中心旋转  $1^\circ$ , 那么在凸多边形旋转一周 (即从  $0^\circ$  到  $360^\circ$ ) 的过程中, 有多长时间凸多边形上所有的点都严格位于两条直线之间 (即位于两条直线之间且不在任意一条直线上) 呢?

### 输入格式

第一行有一个整数  $n$  ( $3 \leq n \leq 10^5$ ), 代表凸多边形的点数。

之后  $n$  行, 每行两个整数  $x, y$  ( $-10^9 \leq x, y \leq 10^9$ ), 代表凸多边形上第  $i$  个点的坐标。坐标按照逆时针顺序给出。数据为非严格凸多边形, 即可能存在三个及以上的点在同一直线上, 但不存在两个坐标相同的点。

第  $n+2$  行包含两个整数  $x_C, y_C$  ( $-10^9 \leq x_C, y_C \leq 10^9$ ), 代表旋转中心。

之后两行, 每行四个整数  $x_A, y_A, x_B, y_B$  ( $-10^9 \leq x_A, y_A, x_B, y_B \leq 10^9$ ), 表示由  $(x_A, y_A), (x_B, y_B)$  两点构成的直线, 保证两点不同, 两条直线平行且不重合。

数据保证旋转中心位于凸多边形内部或边界上且严格位于两条直线之间 (即旋转中心可能在凸多边形的顶点上或边上, 但旋转中心不会出现在两条平行直线的某一条上)。

### 输出格式

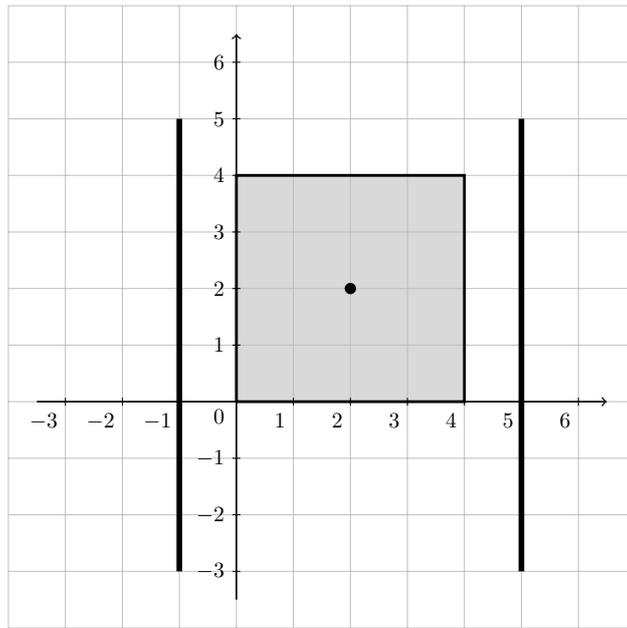
输出一个实数, 表示凸包所有点位于两条直线之间的时间。当且仅当你的答案和正确答案的绝对误差或相对误差不超过  $10^{-6}$  时, 你的答案会被视为正确, 即假设你的答案为  $a$ , 正确答案为  $b$ , 当且仅当  $\frac{|a-b|}{\max(1, |b|)} \leq 10^{-6}$  时你的答案会被视为正确答案。

## 样例

standard input	standard output
4 0 0 4 0 4 4 0 4 2 2 -1 0 -1 1 5 0 5 1	360.0000000000000000
4 0 0 2 0 2 2 0 2 1 1 2 0 2 1 0 0 0 1	0.0000000000000000
3 0 0 -3 -2 3 -2 0 -1 1 0 1 -1 -2 0 -2 -1	41.593143321340491

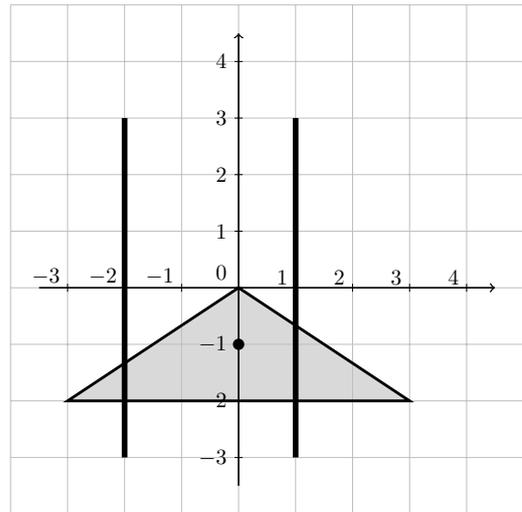
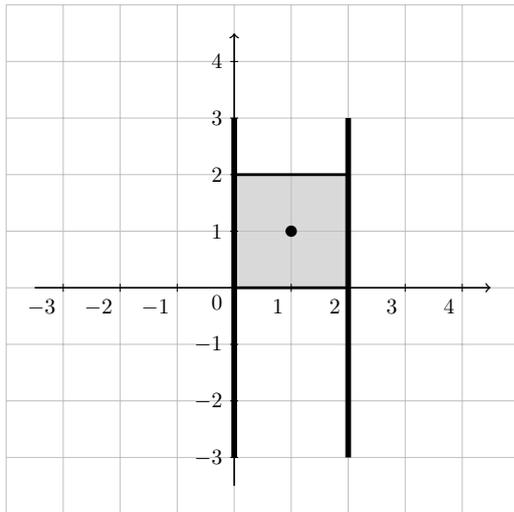
## 提示

第一个样例初始位置如图所示，其中位于 (2,2) 的点为旋转中心。在整个旋转过程中，凸包上的点始终位于两条平行直线之间。



第二个样例初始位置如左图所示, 其中位于  $(1, 1)$  的点为旋转中心。在整个旋转过程中, 凸包上的点或者位于某一条直线上, 或者位于两条直线之间的区域外。

第三个样例初始位置如右图所示, 其中位于  $(0, -1)$  的点为旋转中心。



## Problem E. Serval 的俳句

输入文件: standard input

输出文件: standard output

时间限制: 1 second

空间限制: 512 megabytes

Serval 是加帕里幼儿园的新生。

Serval 在俳句赏析大会上发现了一本神秘书卷，他想从中找出一句俳句。

具体来说，神秘书卷是一个仅包含小写英文字母的字符串  $S$ ，你需要找到满足下列条件的  $S$  的一个子序列  $S'$  作为一句俳句：

- $S'$  的长度  $|S'|$  恰好为 17；
- $S'_1, S'_2, S'_3, S'_4, S'_5$  为同一个字符；
- $S'_6, S'_7, \dots, S'_{11}, S'_{12}$  为同一个字符；
- $S'_{13}, S'_{14}, S'_{15}, S'_{16}, S'_{17}$  为同一个字符。

如果满足条件的子序列存在，则输出这个子序列。若存在多个满足条件的子序列，输出任意一个均可。

如果不存在满足条件的子序列，则输出 none。

我们称  $S'$  是  $S$  的子序列，当且仅当  $S'$  可以从  $S$  中删去任意数量的字符得到。注意  $S'$  的前 5 个字符、中间 7 个字符以及后 5 个字符可以为同一个字符，例如 aaaaaaaaaaaaaaaaaa, bbbbbbccccccbbbbb, dddddeeeeeeeeeeee 都是满足条件的。

### 输入格式

第一行，一个正整数  $|S|$  ( $1 \leq |S| \leq 10^6$ )，表示字符串  $S$  的长度。

第二行，一个长度为  $|S|$  的仅包含小写字母的字符串  $S$ 。

### 输出格式

共一行，如果满足条件的子序列存在，则输出这个子序列，否则输出 none。

### 样例

standard input	standard output
11 lifeispiano	none
22 aaabaaccddccbcccaadaaa	aaaaaccccccaaaaa

## Problem F. 集合之和

输入文件: standard input

输出文件: standard output

时间限制: 1 second

空间限制: 512 megabytes

对于任意有限数集  $A, B$ , 定义二者之和  $A + B$  为:

$$A + B = \{x + y \mid x \in A, y \in B\}$$

例如, 当  $A = \{1, 2\}, B = \{3, 4\}$ , 有  $A + B = \{4, 5, 6\}$ 。注意此处并非可重集, 因此即使  $1 + 4 = 2 + 3 = 5$ ,  $A + B$  也仅包含一个 5。

记有限数集  $A$  中的元素个数为  $|A|$ 。

现给定  $n$ , 试构造数集  $A$  满足  $|A + A| = n$ , 且  $A$  中任一元素  $x$  满足  $0 \leq x \leq 5 \times 10^5$  且  $x$  为整数。

若有多个可行的  $A$ , 输出任意解均可。若  $A$  不存在, 输出  $-1$ 。

### 输入格式

一行, 一个正整数  $n$  ( $1 \leq n \leq 5 \times 10^5$ )。

### 输出格式

若  $A$  存在, 第一行输出一个整数  $|A|$ , 表示集合  $A$  的元素个数。第二行输出  $|A|$  个互不相同的非负整数, 表示  $A$  中的元素。

若  $A$  不存在, 输出  $-1$ 。

### 样例

standard input	standard output
2	-1
3	2 114 514

### 提示

可以证明  $n = 2$  时不存在满足条件的  $A$ 。

当  $A = \{114, 514\}$ , 有  $A + A = \{228, 628, 1028\}$ , 可见  $|A + A| = 3$ 。

## Problem G. Mocha 上大班啦

输入文件: standard input  
 输出文件: standard output  
 时间限制: 1 second  
 空间限制: 512 megabytes

在旭丘幼儿园大班的数学课上, Mocha 学到了位运算和概率。她认为自己已经熟练地掌握了这两个知识点, 于是她找到了同学 Arisa 来出题考考自己。

Arisa 给了 Mocha  $n$  个长度为  $m$  且只包含 0 和 1 的数字串, Arisa 会对这些数字串操作  $q$  次。每次 Arisa 会选择两个数字串  $s_i$  和  $s_j$ , 并选择两个位置  $l, r$ , 对于所有的  $x \in [l, r]$ , 将  $s_j[x]$  替换为  $s_j[x] \& s_i[x]$ ,  $s_j[x]$  为第  $j$  个数字串的第  $x$  位, 其中  $\&$  为位运算中的与运算。但是对于第  $i$  次操作, 只有  $p_i/100$  的概率成功。Arisa 想让 Mocha 计算出  $q$  次操作后,  $n$  个数字串按位与运算后得到的数字串中 1 的个数的期望对 998 244 353 取模的结果。

Mocha 并不能解决这道问题, 但是她不想丢面子, 于是她想请聪明的你帮她计算出这道题的答案。

### 输入格式

第一行两个整数  $n, m$  ( $2 \leq n \leq 1000, 1 \leq m \leq 4000$ ), 代表数字串的个数和长度。

之后  $n$  行每行一个长度为  $m$  且只包含 0 和 1 的数字串。

第  $n+2$  行包含一个整数  $q$  ( $1 \leq q \leq 2 \times 10^5$ ), 代表操作次数。

之后  $q$  行, 每行五个整数  $i, j, l, r, p$  ( $1 \leq i, j \leq n, 1 \leq l \leq r \leq m, 0 \leq p \leq 100$ ), 代表操作两个数字串的编号, 操作的位置范围以及成功概率。保证  $i \neq j$ 。

### 输出格式

输出一个整数, 代表所求期望对 998 244 353 取模的结果。

令  $M = 998\,244\,353$ , 可以证明所求期望可写作既约分数  $\frac{p}{q}$  的形式, 其中  $p, q$  为整数且  $q \not\equiv 0 \pmod{M}$ 。输出的整数应与  $p \cdot q^{-1} \pmod{M}$  相等, 换言之, 输出一个整数  $x$  满足  $0 \leq x < M$  且  $x \cdot q \equiv p \pmod{M}$ 。

### 样例

standard input	standard output
3 3 100 110 111 1 1 2 1 2 0	1

### 提示

由于  $p = 0$ ,  $p/100 = 0$ , 仅有的一次操作不会成功, 所以操作后的数字串始终与原数字串相同。原数字串按位与运算的结果为 100, 仅有一个 1, 所以答案为  $\frac{1}{1} \pmod{998\,244\,353}$ , 即为 1。

作为提醒, 与运算的运算规则是  $0 \& 0 = 0, 0 \& 1 = 0, 1 \& 0 = 0, 1 \& 1 = 1$ 。

## Problem H. 旋转水管

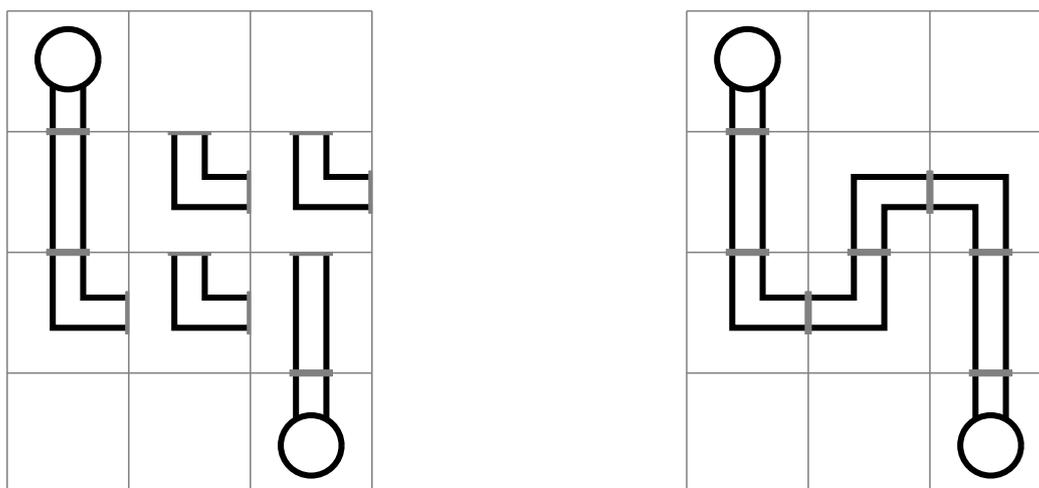
输入文件: standard input  
 输出文件: standard output  
 时间限制: 1 second  
 空间限制: 512 megabytes

金州幼儿园的 Gamer Bellalabella 十分喜欢旋转水管小游戏, 这类游戏的目标是通过旋转游戏中的管道使得水流的输入口与输出口联通。

最近 Bellalabella 发现了一款特殊的旋转水管游戏, 这款游戏中水管只包含 I 形水管与 L 形水管, 并且支持自定义游戏! Bellalabella 立即创造了  $T$  个属于自己的旋转水管游戏:

- 整个游戏构成一个  $4 \times m$  的矩形。
- 第一行仅包含一个位于第  $x$  列的水流输入口, 水流自输入口向下流出。
- 游戏第二、三行有  $m$  列水管, 第  $i$  行  $j$  列为 I 或者 L 表示水管的形状。
- 第四行仅包含一个位于第  $y$  列的水流输出口, 水流输出口仅接受向下流出的水流。

以样例第一个游戏为例, 初始状态如左图所示, 经过旋转  $(2, 2), (2, 3), (3, 2)$  的水管得到如右图所示的状态即可让水流输入口输出口联通。



Bellalabella 想知道自己创造出的  $T$  个旋转水管游戏是否存在一个旋转方案使得输入口与输出口联通, 然而 Bellalabella 的游戏中水管太多了, 自己不能解决, 于是 Bellalabella 将问题交给了正在旁观的你。

### 输入格式

第一行包含一个整数  $T$  ( $1 \leq T \leq 10^5$ ), 表示水管游戏的个数。

每组数据包含三行输入, 第一行包含三个整数  $m, x, y$  ( $1 \leq m \leq 10^5, 1 \leq x, y \leq m$ ), 分别表示矩形的宽度、水流输入口的位置、水流输出口的位置。

第二、三行每行包含  $m$  个字符, 每个字符为大写英文字母 I 和 L 之一, 分别表示 I 形水管与 L 形水管。

保证  $\sum m \leq 5 \times 10^5$ 。

## 输出格式

对于每组输入, 如果可以通过旋转某些水管使得水流可以从规定位置流出, 则输出 YES, 否则输出 NO (大小写不敏感)。

## 样例

standard input	standard output
3	YES
3 1 3	YES
ILL	NO
LLI	
1 1 1	
I	
I	
3 1 3	
IIL	
LLI	

## 提示

对于第一个游戏, 解释如题面所示。

对于第二个游戏, 不需要旋转任何水管。

对于第三个游戏, 无论如何旋转水管都不能联通水流输入输出口。

## Problem I. Oshwiciqwq 的电梯

输入文件: standard input  
输出文件: standard output  
时间限制: 1 second  
空间限制: 512 megabytes

Oshwiciqwq 是一名电梯管理员, 他管理着幼儿园大楼的  $k$  个电梯。这个大楼是一个长方体, 长  $n$  个单位, 宽  $m$  个单位, 高  $h$  个单位。每个长宽高各为 1 个单位的空间是一个房间, 可以用三维坐标  $(x, y, z)$  来表示, 其中  $1 \leq x \leq n, 1 \leq y \leq m, 1 \leq z \leq h$ 。房间里有若干电梯的入口, 可以搭乘电梯前往其他房间。幼儿园大楼有着非常赛博朋克的设计, 有平行三个坐标轴运行的三种电梯, 并且每个电梯是朝坐标增大的方向环线运行的, 在坐标最大处可以运行一次瞬移到坐标最小处。例如, 从  $(x, y, z)$  房间出发, 搭乘沿  $y$  轴运动的电梯, 可以到达  $(x, i, z)$ , 其中  $1 \leq i \leq m$ , 并且可以从  $(x, m, z)$  一步到达  $(x, 1, z)$ 。同时, 不同电梯的运行轨道不相交, 无需担心电梯之间的碰撞。

每个房间都能够搭乘三种电梯各一个, 保证了每两个房间之间都是间接可达的。所有电梯运行一个单位的时间为 1 秒, 容量为无穷大。每个电梯初始所在位置不同, 第  $i$  个电梯 0 秒时在坐标为  $(ex_i, ey_i, ez_i)$  的房间。电梯只能在整秒时刻开始移动或者结束移动, 并且在电梯到达某房间后, 乘客会在到达的同一时刻瞬移进出电梯门。

这些电梯每天需要处理很多乘客的需求。第  $i$  个乘客在  $pt_i$  秒到达, 想要从位于  $(fx_i, fy_i, fz_i)$  的房间到达位于  $(tx_i, ty_i, tz_i)$  的房间。为了更快地到达目的地, 他们会听从电梯管理员的指挥进出电梯。作为高明的电梯管理员, Oshwiciqwq 早已掌握了所有乘客的请求信息, 于是他开始调度电梯的运行和规划乘客的路线。

他会事先把每个乘客的路线计算出来, 并且分段分配给某几个电梯, 每个乘客只会搭乘固定的几个电梯, 并且在规定的房间换乘。他规定乘客的路线为:

$$(fx_i, fy_i, fz_i) \rightarrow (tx_i, fy_i, fz_i) \rightarrow (tx_i, ty_i, fz_i) \rightarrow (tx_i, ty_i, tz_i)$$

即按顺序先后乘坐平行  $x$  轴,  $y$  轴与  $z$  轴的电梯。如果某个阶段无需乘坐电梯已经到达, 则直接忽略。在乘客进行某个阶段的移动时, 只会搭乘对应方向的电梯。

而他给电梯规定的具体运行策略如下:

1. 电梯一开始在给定的坐标, 方向为平行于对应坐标轴的正方向, 一直环线运行。
2. 每到达一个房间, 所有到达当前阶段目的地的乘客会按编号从小到大下电梯。然后在这个房间的乘客会按编号从小到大上电梯。一个乘客下电梯之后, 如果还没有到达最终目的地, 那么会在原地等待下一个电梯, 换乘至少需要一秒的时间。即, 在出电梯的下一秒才能进入另一个电梯。
3. 同一时刻编号小的电梯的所有进出事件发生在编号大的电梯前。

在 Oshwiciqwq 完成了一天的工作之后, 他惊恐地发现电梯管理终端的日志文件损坏了, 无法提交工作报告。此时他手上只有电梯初始情况和乘客的请求记录, 希望你能帮他恢复这个日志文件, 按照时间从小到大列举出乘客进出电梯的信息。

### 输入格式

第一行 3 个正整数  $n, m, h$  ( $2 \leq n, m, h \leq 8$ ), 表示大楼的长、宽、高。

第二行 1 个正整数  $k$  ( $k = n \times m + n \times h + m \times h$ ), 表示电梯的数量。

接下来  $k$  行, 其中第  $i$  行 4 个非负整数  $type_i, ex_i, ey_i, ez_i$  ( $0 \leq type_i \leq 2, 1 \leq ex_i \leq n, 1 \leq ey_i \leq m, 1 \leq ez_i \leq h$ ), 表示编号为  $i$  的电梯种类和初始坐标。  $type_i = 0, 1, 2$  分别表示平行于  $x, y, z$  坐标轴的电梯。保证它们能满足每个房间都能够搭乘三种电梯。

第  $k+3$  行有 1 个正整数  $q$  ( $1 \leq q \leq 50$ ), 表示乘客的数量。

接下来  $q$  行, 其中第  $i$  行 7 个正整数  $pt_i, fx_i, fy_i, fz_i, tx_i, ty_i, tz_i$  ( $1 \leq pt_i \leq 500, 1 \leq fx_i, tx_i \leq n, 1 \leq fy_i, ty_i \leq m, 1 \leq fz_i, tz_i \leq h$ ), 表示编号为  $i$  的乘客到达的时间, 出发房间的坐标以及目的房间的坐标。保证出发房间坐标和目的房间坐标不完全相同。

## 输出格式

输出若干行日志文件。

如果是乘客进入电梯, 输出 “[time] Person person\_id IN Elevator elevator\_id at (x, y, z)”。

如果是乘客走出电梯, 输出 “[time] Person person\_id OUT Elevator elevator\_id at (x, y, z)”。

其中 time 为时间 (秒), 需要加上单位 “s”。elevator\_id 为电梯的编号, person\_id 为乘客的编号,  $(x, y, z)$  为房间的坐标。

## 样例

standard input	standard output
2 2 2	[1s] Person 1 IN Elevator 4 at (2, 2, 2)
12	[2s] Person 1 OUT Elevator 4 at (1, 2, 2)
0 1 1 1	[3s] Person 1 IN Elevator 6 at (1, 2, 2)
0 1 1 2	[4s] Person 2 IN Elevator 2 at (1, 1, 2)
0 1 2 1	[4s] Person 1 OUT Elevator 6 at (1, 1, 2)
0 1 2 2	[5s] Person 2 OUT Elevator 2 at (2, 1, 2)
1 1 1 1	[5s] Person 1 IN Elevator 9 at (1, 1, 2)
1 1 1 2	[6s] Person 2 IN Elevator 8 at (2, 1, 2)
1 2 1 1	[6s] Person 1 OUT Elevator 9 at (1, 1, 1)
1 2 1 2	[7s] Person 2 OUT Elevator 8 at (2, 2, 2)
2 1 1 1	[9s] Person 2 IN Elevator 12 at (2, 2, 2)
2 1 2 1	[10s] Person 2 OUT Elevator 12 at (2, 2, 1)
2 2 1 1	[51s] Person 3 IN Elevator 2 at (2, 1, 2)
2 2 2 1	[52s] Person 3 OUT Elevator 2 at (1, 1, 2)
3	[54s] Person 3 IN Elevator 6 at (1, 1, 2)
1 2 2 2 1 1 1	[55s] Person 3 OUT Elevator 6 at (1, 2, 2)
3 1 1 2 2 2 1	[57s] Person 3 IN Elevator 10 at (1, 2, 2)
50 2 1 2 1 2 1	[58s] Person 3 OUT Elevator 10 at (1, 2, 1)

standard input	standard output
2 3 3	[2s] Person 3 IN Elevator 20 at (2, 2, 1)
21	[3s] Person 3 OUT Elevator 20 at (2, 3, 1)
2 2 1 3	[7s] Person 1 IN Elevator 3 at (1, 1, 2)
0 1 2 3	[7s] Person 2 IN Elevator 3 at (1, 1, 2)
1 1 3 2	[8s] Person 4 IN Elevator 18 at (1, 3, 2)
0 1 2 1	[9s] Person 1 OUT Elevator 3 at (1, 3, 2)
1 2 2 2	[9s] Person 2 OUT Elevator 3 at (1, 3, 2)
2 1 1 3	[9s] Person 5 IN Elevator 16 at (1, 1, 1)
2 1 2 1	[9s] Person 4 OUT Elevator 18 at (2, 3, 2)
1 2 2 3	[10s] Person 4 IN Elevator 5 at (2, 3, 2)
0 1 1 3	[11s] Person 5 OUT Elevator 16 at (1, 3, 1)
2 2 3 2	[12s] Person 4 OUT Elevator 5 at (2, 2, 2)
0 2 2 2	[12s] Person 2 IN Elevator 14 at (1, 3, 2)
0 2 1 2	[13s] Person 2 OUT Elevator 14 at (1, 3, 3)
0 1 1 1	[14s] Person 5 IN Elevator 14 at (1, 3, 1)
2 1 3 2	[15s] Person 5 OUT Elevator 14 at (1, 3, 2)
0 2 3 3	[15s] Person 4 IN Elevator 17 at (2, 2, 2)
1 1 1 1	[17s] Person 4 OUT Elevator 17 at (2, 2, 1)
2 2 2 2	
0 1 3 2	
1 1 2 3	
1 2 3 1	
0 2 3 1	
5	
7 1 1 2 1 3 2	
5 1 1 2 1 3 3	
1 2 2 1 2 3 1	
7 1 3 2 2 2 1	
9 1 1 1 1 3 2	

## 提示

请注意输出的顺序，建议仔细阅读题面中关于电梯运行策略的部分。

## Problem J. Mex Tree

输入文件: standard input

输出文件: standard output

时间限制: 2 seconds

空间限制: 256 megabytes

小水獭幼儿园的同学们都十分擅长图论内容, 作为优等生的 JJLeo 更是如此。

某天, 隔壁金州幼儿园的 Bellalabella 向 JJLeo 请教了这样一个问题。

给定一棵  $n$  个点的树, 编号为  $1, 2, \dots, n$ , 第  $i$  个点的权值为  $v_i$ ,  $v_1, v_2, \dots, v_n$  是  $0, 1, \dots, n-1$  的一个排列。

对于  $k = 0, 1, \dots, n$ , Bellalabella 想找到这棵树的一个非空连通子图, 其点集  $S$  满足  $\text{mex}(\{v_i \mid i \in S\})$  恰好为  $k$ 。如果存在这样的非空连通子图, Bellalabella 想知道点数最多的非空连通子图有多少个点。

其中  $\text{mex}(S)$  表示不属于  $S$  的最小非负整数, 例如:

- $\text{mex}(\{0, 1, 2\}) = 3$ , 因为  $0, 1, 2$  均在  $\{0, 1, 2\}$  中, 而  $3$  是不在集合中的最小非负整数。
- $\text{mex}(\{1, 2\}) = 0$ , 因为  $0$  不在集合中。

JJLeo 虽然精通树的知识, 但是对于  $\text{mex}$  运算却毫无办法, 你能帮他解决这个问题吗?

### 输入格式

第一行, 一个整数  $n$  ( $1 \leq n \leq 10^6$ ), 表示树的结点个数。

第二行,  $n$  个整数  $v_1, v_2, \dots, v_n$ , 表示每个点的权值。保证  $v_1, v_2, \dots, v_n$  是  $0, 1, \dots, n-1$  的一个排列。

若  $n > 1$ , 第三行,  $n-1$  个整数  $f_2, f_3, \dots, f_n$  ( $f_i < i$ ), 表示树上存在边  $(f_i, i)$ 。

### 输出格式

一行以空格分隔的  $n+1$  个整数, 当  $k = i$  ( $i = 0, 1, \dots, n$ ) 时, 若不存在满足条件的非空连通子图, 第  $i+1$  个数为  $0$ , 否则第  $i+1$  个数为满足条件且点数最多的非空连通子图的点数。

### 样例

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

## 提示

对于样例一:

- $k = 0$  时, 不存在非空连通子图的点集  $S$  使得  $\text{mex}(\{v_i \mid i \in S\}) = 0$ 。
- $k = 1$  时, 满足条件的非空连通子图的点集为  $\{1\}$ , 最多有 1 个点。

对于样例二:

- $k = 0$  时, 满足条件的非空连通子图的点集为  $\{2\}$  和  $\{3\}$ , 最多有 1 个点。
- $k = 1$  时, 满足条件的非空连通子图的点集为  $\{1\}$  和  $\{1, 3\}$ , 最多有 2 个点。
- $k = 2$  时, 满足条件的非空连通子图的点集为  $\{1, 2\}$ , 最多有 2 个点。
- $k = 3$  时, 满足条件的非空连通子图的点集为  $\{1, 2, 3\}$ , 最多有 3 个点。

## Problem K. 复合函数

输入文件: standard input

输出文件: standard output

时间限制: 1 second

空间限制: 512 megabytes

给定正整数  $n$ , 并记  $I_n = \{1, 2, \dots, n\}$ 。

给定一个定义在  $I_n$  上的函数  $f$ , 满足对于任意  $x \in I_n$  有  $f(x) \in I_n$ 。

对于任意正整数  $k$ , 当  $k > 1$  时定义  $f^k(x) = f(f^{k-1}(x))$ , 特殊地, 当  $k = 1$  时定义  $f^1(x) = f(x)$ 。

现有  $q$  组询问, 每组询问给定两个正整数  $a, b$ , 试求出有多少个  $x \in I_n$  满足  $f^a(x) = f^b(x)$ 。

### 输入格式

第一行, 一个正整数  $n$  ( $1 \leq n \leq 10^5$ ), 表示  $f$  的定义域的大小。

第二行,  $n$  个正整数  $f(1), f(2), \dots, f(n)$  ( $1 \leq f(x) \leq n$ ), 表示函数  $f$  的取值。

第三行, 一个正整数  $q$  ( $1 \leq q \leq 10^5$ ), 表示询问组数。

接下来  $q$  行, 每行两个正整数  $a, b$  ( $1 \leq a, b \leq 10^{18}$ ), 表示一组询问。

### 输出格式

输出共  $q$  行。对于每组询问, 输出一行, 一个整数, 表示满足  $f^a(x) = f^b(x)$  的  $x$  的个数。

### 样例

standard input	standard output
1 1 1 1 1	1
6 1 1 4 5 1 4 5 1 919810 19 19810 191 9810 1919 810 19198 10	3 6 6 6 6

## Problem L. 串串串串……

输入文件: standard input  
 输出文件: standard output  
 时间限制: 10 seconds  
 空间限制: 512 megabytes

加帕里幼儿园的新生 Serval、旭丘幼儿园的新生 Mocha、金州幼儿园的新生 Bellalabella 和小水獭幼儿园的新生 JJLeo 各自想了一个字符串, 幼儿教育局局长 Toxel 把这四个字符串缝合到一起, 同时从各个次元的幼儿园征集了一些字符串, 并由此衍生了一个计数问题, 你能帮他解决这个问题吗?

具体来说, 给定  $n$  个字符串  $s_1, s_2, \dots, s_n$ , 和一个正整数  $m$ 。定义一个非空字符串元组  $(T_1, T_2, \dots, T_k)$  是字符串  $t$  的合法划分当且仅当:

- 将  $T_1, T_2, \dots, T_k$  按顺序拼接起来得到的字符串和  $t$  相同。
- 对任意  $1 \leq i \leq k$ ,  $T_i$  是  $s_1, s_2, \dots, s_n$  中至少  $m$  个字符串的子串。

其中  $k$  称为这个合法划分的长度。

定义  $f(t)$  为字符串  $t$  所有合法划分中的最短长度。特别地, 如果不存在  $t$  的合法划分,  $f(t) = 0$ 。

给定正整数  $l$  和  $r$ , 设所有长度不小于  $l$  也不大于  $r$  的字符串所组成的集合为  $S$ , 请你求解:

$$\left( \sum_{t \in S} f(t) \right) \bmod 998\,244\,353$$

本题中, 字符集大小为  $10^9$ , 使用正整数  $1, 2, \dots, 10^9$  来代表每个字符。

字符串  $a$  是字符串  $b$  的子串当且仅当  $a$  可以通过删去  $b$  的一个前缀 (可以为空) 和一个后缀 (可以为空) 得到。

### 输入格式

第一行, 四个正整数  $n, m, l, r$  ( $1 \leq m \leq n \leq 5000, 1 \leq l \leq r \leq 10^{18}$ ), 含义同题目描述。

接下来  $n$  行, 第  $i+1$  行第一个数  $|s_i|$  ( $1 \leq |s_i| \leq 5000$ ), 表示字符串  $s_i$  的长度。接下来  $|s_i|$  个数, 第  $j$  ( $1 \leq j \leq |s_i|$ ) 个数  $s_{i,j}$  ( $1 \leq s_{i,j} \leq 10^9$ ) 表示  $s_i$  的第  $j$  个字符。

保证  $\sum_{i=1}^n |s_i| \leq 5000$ 。

### 输出格式

共一行, 一个非负整数, 表示答案。

## 样例

standard input	standard output
2 1 1 2 1 1 2 2 1	9
3 3 1 5 4 1000000000 1 1 1 4 998244353 1 1 2 4 999244353 1 1 3	9
5 3 1 1000000000 10 1 2 3 4 5 6 7 8 9 10 10 2 3 4 5 6 7 8 9 10 11 10 3 4 5 6 7 8 9 10 11 12 10 4 5 6 7 8 9 10 11 12 13 10 5 6 7 8 9 10 11 12 13 14	387621546

## 提示

我们用  $[T_1, T_2, \dots, T_{|t|}]$  表示字符串  $t$ , 其中  $T_i$  ( $1 \leq i \leq |t|$ ) 表示  $t$  的第  $i$  个字符。

第一个样例中, 对所有  $f(t) > 0$  的字符串  $t$  列举如下:

- $f([1]) = 1$
- $f([2]) = 1$
- $f([1, 1]) = 2$
- $f([1, 2]) = 2$
- $f([2, 1]) = 1$
- $f([2, 2]) = 2$

第二个样例中, 对所有  $f(t) > 0$  的字符串  $t$  列举如下:

- $f([1]) = 1$
- $f([1, 1]) = 1$
- $f([1, 1, 1]) = 2$
- $f([1, 1, 1, 1]) = 2$
- $f([1, 1, 1, 1, 1]) = 3$