NOIP2024 模拟赛

题目名称	说唱入门教学	画画入门教学	恋爱入门教学	种花入门教学
题目类型	传统型	传统型	传统型	传统型
可执行文件名	rap.exe	draw.exe	love.exe	plant.exe
输入文件名	rap.in	draw.in	love.in	plant.in
输出文件名	rap.out	draw.out	love.out	plant.out
每个测试点时限	1.0 秒	1.0 秒	2.5 秒	1.0 秒
内存限制	1024 MiB	512 MiB	512 MiB	512 MiB
测试点数目	21	20	20	6
测试点是否等分	否	是	是	否

提交源程序文件名

对于 C++ 语言	rap.cpp	draw.cpp	love.cpp	plant.cpp

编译选项

对于 C++ 语言	-O2 -std=c++14 -Wl,--stack=1073741824

注意事项

1. 选手提交的源程序请直接存放在个人目录下，无需建立子文件夹。
2. 文件名（包括程序名和输入输出文件名）必须使用英文小写。
3. C++ 中函数 main() 的返回值必须是 int，值必须为 0。
4. 若无特殊说明，结果比较方式为**忽略行末空格、文末回车后的全文比较**。
5. 程序可使用的栈空间大小与该题内存空间限制一致。
6. 使用 g++ 同一目录下的 size 可以查看程序静态空间大小。
7. 若无特殊说明，每道题的**代码大小限制为 100KB**。
8. 若无特殊说明，输入与输出中同一行的相邻整数、字符串等均使用一个空格分隔。
9. 输入文件中可能存在行末空格，请选手使用更完善的读入方式（例如 scanf 函数）避免出错。
10. 直接复制 PDF 题面中的多行样例，数据将带有行号，建议选手直接使用对应目录下的样例文件进行测试。
11. 使用 std::deque 等 STL 容器时，请注意其内存空间消耗。

12. 请务必使用题面中规定的的编译参数，保证你的程序在本机能够通过编译。此外**不允许在程序中手动开启其他编译选项**，一经发现，本题成绩以 0 分处理。
13. 评测时采用的机器配置为 `12th Gen Intel(R) Core(TM) i7-12700 2.10 GHz CPU`，内存 `16GB`，上述时限以此配置为准。
14. 评测在 `windows` 下使用 `LemonLine` 进行，编译器为 `TDM-GCC 10.3.0-2`，请注意**环境差异**。
15. **若无特殊说明，评测默认不开启捆绑测试。**

说唱入门教学（rap）

【题目背景】

JF 是一个热爱说唱的男高，他立志~~泡到全世界的妹子~~让全世界的人都听到他的音乐。在这之前他对中国说唱歌手的歌曲进行研究，并作出类似下图的表格。

他想研究完所有市面上的说唱歌手的歌曲，但这样手动计算不现实，你能帮帮他吗？

【题目描述】

根据小学知识，一个**拼音**由**声母**和**韵母**组成。

在本题中，规定声母有

```
b,p,m,f,d,t,n,l,g,k,h,j,q,x,zh,ch,sh,r,z,c,s,y,w
```

韵母有

```
a,o,e,i,u,ai,ei,ui,ao,ou,er,an,en,in,un,ang,eng
ing,ong,ia,ua,ie,ue,iu,ian,iang,uan,iong,uo,iao,uang
```

声母和韵母任意组合得到一个拼音。例如「原神」一词的拼音是 `yuan shen`，「我爱嘉然」一句的拼音是 `wo ai jia ran`。本题中，组合得到的拼音在现实中**不一定真实存在**，例如拼音 `fiong`，但我们认为它是一个合法的拼音。

称两个拼音**押韵**，当且仅当他们的**韵母**相同。

特别地，对于韵母为 `ia,ua,ie,ue,ian,iang,uan,iong,uo,uang` 的拼音，在比对韵母的时候，忽略韵母最前面一位的 `i` 或 `u`。例如，韵母为 `ian,uan` 的拼音和韵母为 `an` 拼音押韵，韵母为 `ia,ua` 的拼音和韵母为 `a` 的拼音押韵。

特别地，前鼻音和后鼻音押韵，也就是说，韵母为 `an` 的拼音和韵母为 `ang` 的拼音押韵，韵母为 `en` 的拼音和韵母为 `eng` 的拼音押韵，韵母为 `in` 的拼音和韵母为 `ing` 的拼音押韵。

综合上述两点，若两个拼音的韵母为 `ian,iang,an,ang,uan,uang` 中的任意两个，则这两个拼音押韵。

若两句话末尾的 1 个拼音是押韵的，则称这两句话是一个**单押**。例如，分别以拼音 `ma` 和 `ba` 结尾的两句话是一个单押。

若两句话末尾的 2 个拼音是押韵的，称称这两句话是一个**双押**。例如，分别以拼音 `sheng yin` 和 `geng xin` 结尾的两句话是一个双押。

类似地，我们可以我们定义**三押**，**四押**，$\cdots$，k **押**。例如，分别以拼音 `heng da ji hua` 和 `meng na li sha` 的两句话是一个四押。

现在，给定共有 n 句歌词且每句歌词长度（即包含的**拼音**个数）不超过 m 的歌的全部歌词，请对**全部** $\frac{n(n-1)}{2}$ **对句子**进行比对判断押韵，并分别求出这首歌单押，双押，$\cdots$，m 押的个数。需要注意的是，在计数时，若两句话是一个 k 押，虽然它们同时是一个单押，双押，$\cdots$，$k-1$ 押，但这个 k 押**不在其中计数**。

特别地，当作者名字为 `Wiz_H` 时，因为「不是吧，不是吧，难道单押也算押」，所以在进行计数时，**不计算单押数，输出时单押个数视为 0**；

当作者名字为 `BeiBei` 时，因为「他只有九指半」，所以在进行计数时，**不计算满足 $k>9$ 的 k 押个数，输出时这些 k 押的个数视为 0**；

当作者名字为 `Kris_Wu` 时，因为他歌曲惨遭下架，所以**什么都不要输出**。

【输入格式】

从文件 ***rap.in*** 中读入数据。

第一行两个正整数 n, m，分别代表歌曲总句子数量，以及最长句子的长度。**请注意，句子长度指拼音个数而不是字符串长度**。

第二行 n 个正整数 $a_1, a_2, \cdots, a_n$，其中 a_i 代表第 i 个句子的长度。

第三行一个字符串，表示歌曲名字。**表示歌曲名字的字符串可能会包含空格。**

第四行一个字符串，表示作者名字。下文中**子任务**一栏列出了每个测试点歌词对应的作者名字。

接下来 n 行，第 i 行 a_i 个字符串 $p_{i,1}, p_{i,2}, \cdots, p_{i,a_i}$，其中 $p_{i,j}$ 表示第 i 句歌词中的第 j 个拼音。**不保证**测试数据中两个拼音之间只有一个空格。

【输出格式】

输出到文件 ***rap.out*** 中。

输出 m 个非负整数，依次表示这首歌单押，双押，$\cdots$，m 押的个数。特别地，当作者名字为 `Kris_Wu` 时，什么都不要输出。

【样例 1 输入】

```
4 7
5 5 7 5
shu ya zi
HF
men qian da qiao xia
```

```
you guo yi qun ya
kuai lai kuai lai shu yi sha
er si liu qi ba
```

【样例 1 输出】

```
5 1 0 0 0 0 0
```

【样例 1 解释】

第三句 `kuai lai kuai lai shu yi sha` 和第四句 `er si liu qi ba` 是一个双押，剩余任意两句都是单押。

【样例 2】

见选手目录下的 ***rap/rap2.in*** 与 ***rap/rap2.ans***。

【样例 3】

见选手目录下的 ***rap/rap3.in*** 与 ***rap/rap3.ans***。

【样例 4】

见选手目录下的 ***rap/rap4.in*** 与 ***rap/rap4.ans***。

【样例 5】

见选手目录下的 ***rap/rap5.in*** 与 ***rap/rap5.ans***。

【子任务】

对于 100% 的数据，$1 \leq n, m \leq 3000$。
注意作者名字中的下划线为单个 _。

测试点	$n \leq$	$m \leq$	作者	分值
1	2	10	jeefy	4
2	10	20	HF	5
3	10	20	GAI	5
4	500	500	JYS	5
5	500	500	Wiz_H	5
6	500	500	Masiwei	5
7	500	500	D_Shine	5
8	500	500	BeiBei	5
9	500	500	Pharaoh	5
10	500	500	PGone	5
11	2000	2000	BeiBei	5
12	2000	2000	404	5
13	2000	2000	Te_Lt_Cy	5
14	2000	2000	ffgg	5
15	2000	2000	Young	5
16	2000	2000	Ricky	5
17	2000	2000	Jony_J	5
18	2000	2000	Rairn	5
19	3000	3000	BeiBei	5
20	3000	3000	YZH	5
21	3000	3000	Kris_Wu	1

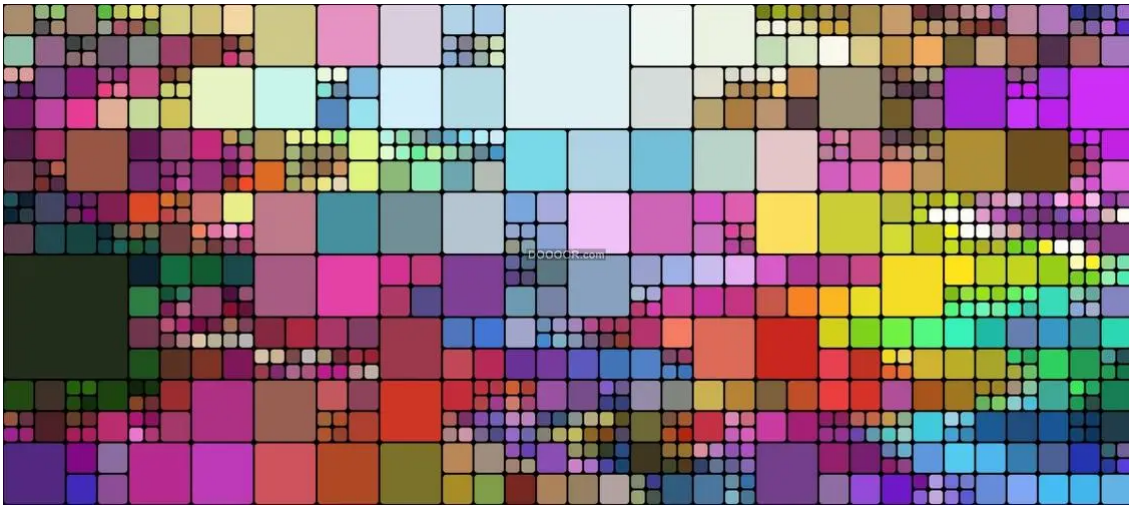
【提示】

由于输入量较大，建议使用更优秀的输入方式。

画画入门教学（draw）

【题目背景】

JH 是一个热爱画画的高中生，他最近沉迷于珂爱的矩阵无法自拔，想要画一个像下面这样的漂亮矩阵：



可惜，因为他的画图工具出锅了，所以需要你的帮 忙，当然他不会为难你画出像上面这样复杂的图啦！

他只是让你帮他做一点前置工作而已（顺便让你的画画水平得到提高），当然啦他也会尽快把他的工具修好，但是可能需要一天时间 ~~一天时间？当然是故意的啦！嘻嘻！~~

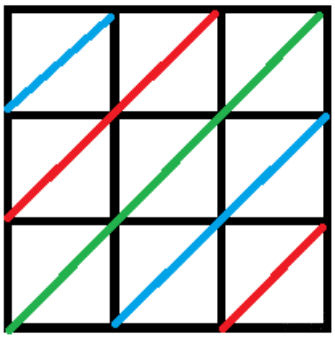
【题目描述】

JH 给你了一个正整数 n，需要你帮助他将一个 $n \times n$ 的方阵 S 填上数。为了保证美观，方阵 S 需要满足以下性质：

- 它的每一行构成一个 $1 \sim n$ 的排列。
- 它的每一条左下角向右上角的**斜线**构成一个 $1 \sim n$ 的排列。若斜线从矩阵上方超出，则在该列最下方继续向右上方延伸，若从矩阵右方超出，则在该行最左侧继续延伸，直到超出第 n 列。因此，起点为 $S_{i,1}$ 的斜线，包含每一列的元素恰好一个，且对于满足 $1 \leq k \leq n$ 的 k，起点为 $S_{i,1}$ 的斜线包含元素 $S_{(i-k) \bmod n+1,k}$，但不包含第 k 列的其它元素。
- 称起点为 $S_{n,1}$ 的斜线为**对角线**。方阵 S 需要满足，存在一个 $n \times n$ 的方阵 T，使得：将 S **对角线**上的数进行任意重排（**可以不改变顺序**）后能够得到 T，且 T 的每一列可以构成一个 $1 \sim n$ 的排列。

如下图。图中三条颜色的线为三条斜线，特别地，绿色的线为对角线。

请你求出一个满足上述条件的方阵 S，以及与 S 对应的方阵 T。可以证明，解总是存在的。



【输入格式】

从文件 ***draw.in*** 中读入数据。
一行一个正整数 n。

【输出格式】

输出到文件 ***draw.out*** 中。
前 n 行，第 i 行 n 个正整数 $S_{i,1}, S_{i,2}, \cdots, S_{i,n}$。
接下来 n 行，第 i 行 n 个正整数 $T_{i,1}, T_{i,2}, \cdots, T_{i,n}$。
如果存在多种解，输出任意一种均可。

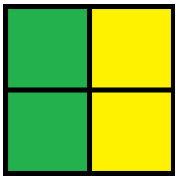
【样例 1 输入】

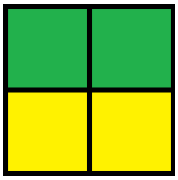
```
2
```

【样例 1 输出】

```
1 2
1 2
1 1
2 2
```

【样例 1 解释】





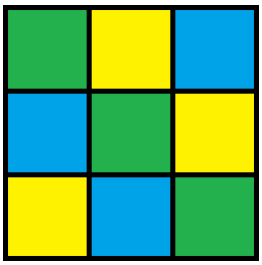
【样例 2 输入】

```
3
```

【样例 2 输出】

```
1 2 3
3 1 2
2 3 1
1 2 3
3 1 2
2 3 1
```

【样例 2 解释】



【子任务】

对于 20% 的数据，$n \bmod 2 = 1$。
对于其余 80% 的数据，$n \bmod 2 = 0$。
对于 100% 的数据，$1 \leq n \leq 10^3$。

恋爱入门教学（love）

【题目背景】

JF 学会了说唱，夺得了中国第一男高的称号，~~迷倒子万千迷妹~~收获了大笔的金钱。他~~广撒网~~广交友，交到了许多卡哇伊的 npy，可是麻烦随之而来。

【题目描述】

每一个 npy 对 **JF** 都有一个好感度 $Favorbility_i$（下文简记为 F_i）。而 **JF** 秉持着公平的原则，对于每一个 npy 的好感度 f 是一样的。于是，好感的**不对等**会造成一定的麻烦。

每一个 npy 都有一个麻烦率 $Troublesome_i$（下文简记为 T_i），如果纠缠着 **JF**，那么 T_i 为正，如果是冷漠，则 T_i 为负。或许可能是真爱，有的 npy 的 T_i 为 0。

但是与每一个 npy 相处会有一个基础的麻烦度 B_i，由于也可能是帮助，所以 B_i 可能小于零。

JF 为了不让自己被群殴，所以想要**最小化**自己的麻烦度。但是 **JF** 脑袋要炸掉了，所以求上了 **JH** 寻求帮助。不过 **JH** 认为所有 npy 都是麻烦的，因为这只会影响他画画的效率。所以他会对所有的麻烦度取个**绝对值**。

JF 有时会结交到新的 npy，但也可能会失去一些 npy，所以 **JH** 需要不断的帮助 **JF** 完成这个任务。可 **JH** 觉得每次都要全部算一次太麻烦了，所以希望你能帮助他快速的完成这个任务。

形式化来说，假定当前有 k 个 npy，分别对应一个三元组 (F_i, T_i, B_i)，最小化：

$$\sum_{i=1}^{k} |T_i(F_i - f) + B_i|$$

【输入格式】

从文件 ***love.in*** 中读入数据。

第一行一个数 n。

接下来 n 行，每行第一个数 op 表示 **JF** 的交友情况：

- 如果 $op = 1$，那么接下来跟着三个整数 (F_i, T_i, B_i)，表示 **JF** 新交了一个 npy。
- 如果 $op = 2$，那么接下来跟着一个数 k，表示 **JF** 失去了第 k 个交到的 npy，保证这个 npy 之前一定对 **JF** 忠心不二（没有分手）。

【输出格式】

输出到文件 ***love.out*** 中。

在每交到或者失去一个 npy 之后，你都要输出 **JF** 当前最小的麻烦度。

每行一个数，一共 n 行，误差在 10^{-5} 以内则视为正确。

【样例 1 输入】

```
3
1 1 2 3
1 4 5 6
2 1
```

【样例 1 输出】

```
0
5.4
0
```

【样例 2】

见选手目录下的 ***love/love2.in*** 与 ***love/love2.ans***。

【样例 3】

见选手目录下的 ***love/love3.in*** 与 ***love/love3.ans***。

【样例 4】

见选手目录下的 ***love/love4.in*** 与 ***love/love4.ans***。

【子任务】

对于 100% 的数据 $1 \leq n \leq 5 \times 10^5, |F_i|, |T_i|, |B_i| \leq 10^4$。
特殊性质：

- A：保证 $T_i = 1$
- B：保证没有删除操作

测试点编号	$n \leq$	特殊性质
$1 \sim 3$	10	无
4		A
5		B
$6 \sim 8$	5000	无
9		A
10		A, B
$11 \sim 14$	100000	无
15		B
$16 \sim 20$	500000	无

种花入门教学（plant）

【题目背景】

JH 画完画之后喜欢种花。

【题目描述】

他一共种了 n 盆花，编号为 $1 \sim n$，这些花的初始高度都为 0。**JH** 还有一个长度为 n 的正整数序列 a，**JH** 认为这 n 盆花是美观的当且仅当第 i 盆花的高度为 a_i。

为了让这些花变美观，**JH** 会进行若干次修剪/浇水操作。一次修剪/浇水操作定义为使一盆花的高度 $-1/+1$。由 **JH** 人工进行修剪/浇水操作每次都会花费 1 的代价。由于 **JH** 觉得这样太麻烦了，他还发明了一台种花自动机。这台机器可以花费 c 的代价对于一个区间统一进行一次修剪/浇水操作。**JH** 绝顶聪明，他一定会选择花费代价最少的方式让花变美观。

由于 **JH** 的心情依赖于画画的结果，因此对于每一个 $1 \leq i \leq n$ 的 i，a_i 会在一个大小为 m 的可重集合 S_i 里等概率随机选取。**JH** 想知道，他让这些花变美观需要的期望代价是多少。

【输入格式】

从文件 ***plant.in*** 中读入数据。

第一行 3 个正整数 n, m, c。

接下来 n 行，第 i 行 m 个整数，描述了集合 S_i。

【输出格式】

输出到文件 ***plant.out*** 中。

一个数表示期望代价对 $10^9 + 7$ 取模的结果。

【样例 1 输入】

```
8 1 2
8
4
5
9
4
2
10
```

```
1
```

【样例 1 输出】

```
26
```

【样例 2 输入】

```
4 2 2
2 2
9 1
5 2
2 3
```

【样例 2 输出】

```
875000016
```

【子任务】

定义 mx_i 为 S_i 中的最大元素。

本题开启捆绑测试

子任务编号	n	m	c	a_i	mx_i	分数
1	$=1$	$=1$	$\leq n$	$\leq 10^9$	$\leq 10^9$	1
2	$=3$	$=2$		≤ 10	≤ 10	2
3	$=4$					3
4	≤ 100	$=1$		≤ 100	≤ 100	24
5	$\leq 10^6$			$\leq 10^9$	$\leq 10^9$	36
6	≤ 100	≤ 70				34