NOI2022 省选

DAY1

时间：2022 年 4 月 16 日 08:30 ～ 13:00

题目名称	预处理器	填树	学术社区
题目类型	传统型	传统型	传统型
目录	`preprocessor`	`tree`	`community`
可执行文件名	`preprocessor`	`tree`	`community`
输入文件名	`preprocessor.in`	`tree.in`	`community.in`
输出文件名	`preprocessor.out`	`tree.out`	`community.out`
每个测试点时限	1.0 秒	2.0 秒	1.5 秒
内存限制	512 MiB	512 MiB	512 MiB
测试点数目	10	10	25
测试点是否等分	是	是	是

提交源程序文件名

对于 C++　语言	`preprocessor.cpp`	`tree.cpp`	`community.cpp`

编译选项

对于 C++　语言	`-O2 -std=c++14`

注意事项（请仔细阅读）

1. 文件名（程序名和输入输出文件名）必须使用英文小写。
2. C/C++ 中函数 main() 的返回值类型必须是 int，程序正常结束时的返回值必须是 0。
3. 提交的程序代码文件的放置位置请参考各省的具体要求。
4. 因违反以上三点而出现的错误或问题，申诉时一律不予受理。
5. 若无特殊说明，结果的比较方式为全文比较（过滤行末空格及文末回车）。
6. 选手提交的程序源文件必须不大于 100KB。
7. 程序可使用的栈空间内存限制与题目的内存限制一致。
8. 全国统一评测时采用的机器配置为：Inter(R) Core(TM) i7-8700K CPU @3.70GHz，内存 32GB。上述时限以此配置为准。
9. 只提供 Linux 格式附加样例文件。
10. 评测在当前最新公布的 NOI Linux 下进行，各语言的编译器版本以此为准。

预处理器（preprocessor）

【题目描述】

宏是 C/C++ 语言的一项特性，它根据预先定义的规则进行文本替换（也被称为“宏展开”），能够实现定义常量、简化代码重复输入等功能。例如：

```
#define PI 3.14159
double area = PI * r * r;
```

以上代码经过宏展开后变为：

```
double area = 3.14159 * r * r;
```

其中，宏定义命令变成了空行，而其他行中的宏被展开成了规则定义的文本。

C/C++ 语言代码在编译时对宏的处理由**预处理器**完成。你的任务是实现一个简化版的预处理器，要求如下：

- 代码由**行**组成，每行除行末的换行符外，均由可打印 ASCII 字符（ASCII 码范围 32—126）组成。每行要么是**预处理命令**（以 # 开头），要么是**普通文本**（其他情况）。
- 预处理器逐行处理代码，
  - 如果是预处理命令，执行该命令，并输出一个空行。
  - 如果是普通文本，对其进行宏展开并输出结果。
- 预处理命令有两种，分别是宏定义命令 `#define` 和取消宏定义命令 `#undef`。
  - 宏定义命令的格式为 `#define <name> <content>`,其中第一部分 `#define` 是命令名，第二部分 `<name>` 是要定义的宏的名字，第三部分 `<content>` 是要定义的宏的展开内容。
  - 取消宏定义命令的格式为 `#undef <name>`，其中第一部分 `#undef` 是命令名，第二部分 `<name>` 是要取消的宏的名字。

  以上两种预处理命令中，相邻两部分之间都严格用一个空格分隔。`<name>` 是由大小写字母和数字以及下划线组成的**标识符**（一个或多个字符），`<content>` 可以包含任意可打印 ASCII 字符（零个或多个字符）。一个宏定义的**有效范围**是从它定义所在行开始到后续最近的宏名匹配的取消定义所在行为止（如果没有对应的取消定义，则有效范围一直覆盖到文件结束）。

对普通文本进行宏展开时，将一行文本中每段**连续极长**的大小写字母和数字以及下划线视为标识符（而不是其中一部分），其余为**其他字符**。从左到右依次对文本中的标识符进行宏展开：

1. 如果该标识符是有效的宏名，则用对应的展开内容替换它，此时该宏名进入正在展开的状态，直到本流程结束；否则原样保留宏名。例如，若宏 `A` 定义为 `b`，则

文本 A 展开结果为 b（发生替换），文本 B 展开结果仍然为 B（未定义，不替换），文本 AA 展开结果仍然为 AA（AA 是不同于 A 的另一个标识符，未定义），而文本 A*B 开展结果为 b*B。

2. 替换发生后，如果展开内容中包含标识符，重复应用以上的展开操作，称为“多次展开”。例如，若宏 A 定义为 B，宏 B 定义为 c，则文本 A 的展开结果为 c。
3. 如果待展开的宏名与正在进行展开的某个宏名相同，称为“递归展开”，此时该宏名不再展开。本规则用来防止无限递归展开。例如，若宏 A 定义为 B+a，宏 B 定义为 A+b，则文本 A 展开结果为 A+b+a，由于最初的 A 处于正在展开状态，因此 A+b+a 里的 A 不再展开。
4. 其他字符原样保留。

注意：出于简化的目的，本题的要求与 C/C++ 语言标准里的描述不完全一致，请以上面的要求为准。最明显的区别是本题只有标识符和其他字符两类词法单元，没有数值、字符串、注释等。

【输入格式】

从文件 ***preprocessor.in*** 中读入数据。

输入的第一行包含一个正整数 n，表示要处理的代码行数。

接下来的 n 行是要处理的代码。

【输出格式】

输出到文件 ***preprocessor.out*** 中。

输出 n 行，为输入逐行预处理后的结果。

【样例 1 输入】

```
5
#define BEGIN {
#define END }
#define INTEGER int
class C BEGIN INTEGER x; END;
INTEGER main() BEGIN C c; END
```

【样例 1 输出】

```

```

```
class C { int x; };
int main() { C c; }
```

【样例 2】

见选手目录下的 ***preprocessor/preprocessor2.in*** 与 ***preprocessor/preprocessor2.ans***。

【样例 3】

见选手目录下的 ***preprocessor/preprocessor3.in*** 与 ***preprocessor/preprocessor3.ans***。

【数据范围】

对 20% 的数据，不会出现宏定义命令 `#define` 和宏取消定义命令 `#undef`。

对另外 20% 的数据，不会出现多次展开的情况，且不会出现宏取消定义命令 `#undef`。

对另外 20% 的数据，不会出现多次展开的情况。

对另外 20% 的数据，不会出现递归展开的情况。

对其余数据，无特殊限制。

对 100% 的数据，$n \leq 100$，输入的每行字符数都不超过 100，且保证输出的每行字符数都不超过 1 000（字符数均不计行末换行符）。保证输入数据中的预处理命令都是合法的，包含但不限于：

- `#` 字符只会出现在预处理命令所在行的第一个字符的位置，其他任何位置（包括预处理命令和普通文本）都不会出现 `#` 字符。
- 宏定义和取消定义命令的格式是正确的，严格遵循题面所描述的格式。
- 同一个宏在取消定义之前不会被再次定义。
- 要取消定义的宏在之前被定义过且还没有被取消过。

也就是说，**你不需要做任何语法和语义的错误检查**。

【提示】

本题进行输入时建议使用 C++ 语言的按行读入字符串功能，示例如下：

```
#include <iostream>
#include <string>
using namespace std;
string line;
// 从 cin 读入一行，放入 line 中（换行符被舍弃）
```

```
getline(cin, line);
```

也可以使用 C 语言提供的 fgets 函数，示例如下：

```
#include <stdio.h>
#define MAX_LEN 200
char line[MAX_LEN];
// 从 stdin 读入一行，放入 line 中（包含换行符）
fgets(line, MAX_LEN, stdin);
```

注意：在读取行数 n 之后可能需要额外读取一行以忽略其后的换行符。

填树（tree）

【题目描述】

有一棵 n 个节点的无根树，刚开始树上每个节点的权值均为 0。KK 想对这棵树进行一些修改，他会任选一个节点作为初始的当前节点，然后重复以下动作：

1. 将当前节点 i 的权值修改为一个**正整数** x，需满足 $l_i \le x \le r_i$。其中 l_i, r_i 是输入中给出的两个正整数。
2. 结束修改过程，或移动到一个与当前节点相邻的权值为 0 的节点（如果不存在这样的节点，则必须结束修改过程）。

现在 KK 有两个问题：

1. 在修改结束后，可以得到多少棵不同的树，满足树上**非零权值**的最大值和最小值的差小于等于 K？其中 K 是输入中给出的一个正整数。
2. 这些满足条件的树的权值之和为多少？（树的权值定义为这棵树上所有节点的权值之和）

你需要输出这两个问题的答案模 $10^9 + 7$。我们认为两棵树不同当且仅当至少存在一个节点的权值不同。

温馨提示：

1. KK 至少会修改一个节点（初始节点）。
2. 实质上 KK 会修改树上的任意一条路径，最后需要满足这条路径上的点的权值最大值和最小值之差小于等于 K。

【输入格式】

从文件 ***tree.in*** 中读入数据。

第一行两个正整数 n, K，表示节点数和权值差的最大值。

接下来 n 行，每行两个正整数 l_i, r_i，表示第 i 个节点修改后权值的最小值和最大值。

接下来 $n-1$ 行，每行两个正整数 u_i, v_i，表示节点 u_i 和 v_i 之间有一条边。数据保证形成一棵树。

【输出格式】

输出到文件 ***tree.out*** 中。

输出两行，每行一个整数，分别表示第一问和第二问的答案模 $10^9 + 7$ 的值。注意，如果你不打算回答第二问，请在第二行任意输出一个整数。如果输出文件只有一行，则会因格式不符合要求被判 0 分。

【样例 1 输入】

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

【样例 1 输出】

```
14
78
```

【样例 1 解释】

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
节点 1	2	3	2	3	3	3	3	3	0	0	0	0	0	0
节点 2	0	0	3	3	4	0	4	3	3	4	5	0	0	0
节点 3	0	0	0	0	0	4	4	4	0	0	0	4	5	6

表格中列出了全部 14 棵满足条件的树，将这些树的权值加起来为 78。

【样例 2】

见选手目录下的 ***tree/tree2.in*** 与 ***tree/tree2.ans***。

【样例 3】

见选手目录下的 ***tree/tree3.in*** 与 ***tree/tree3.ans***。

【数据范围】

对于 100% 的数据，$1 \leq n \leq 200,\ 1 \leq l_i \leq r_i \leq 10^9, 1 \leq K \leq 10^9$。

测试点	$n \leq$	$r_i, K \leq$	其他限制
1	5	10	无
$2 \sim 3$	30	10^9	
4		500	
$5 \sim 6$	200	200000	
$7 \sim 8$		10^9	A
$9 \sim 10$			无

特殊限制 A：所有点构成一条链, 编号为 i 的点和编号为 $i+1$ 的点之间有连边

【评分方式】

本题共 10 个测试点，每个测试点 10 分。其中回答正确第一问可得 7 分，回答正确第二问可得 3 分。

学术社区（community）

小 I 的温馨提示：在题目描述中有形式化的题面，你可以选择跳过题目背景。同时请仔细将本题的除题目背景以外的所有内容进行完整阅读后再进行做题。

【题目背景】

小 I 是一个喜欢 OI 的选手，不过，与其说小 I 喜欢 OI，不如说小 I 喜欢的是他最经常使用的 OJ——FCCOJ——上的趣味功能：学术社区。虽说名字叫学术社区，但小 I 和网友们能够谈论的东西远不止学术。每天学术社区里总会出现不少吸引小 I 注意的帖子。今天小 I 在学术社区冲浪时发现了一个这样的帖子：

`builtin_clz`：萌新求助，学术社区这题，本机 AC 提交 RE

`builtin_ctz`：`builtin_clz`楼下

`jinkela`：`builtin_ctz`楼下

`builtin_ctz`：`builtin_clz`楼上

`builtin_clz`：能不能别魔怔了，大家正经回答问题

`OrzTourist`：`builtin_clz`楼下

`OrzTourist`：`OrzTourist`楼下

`builtin_clz`：怎么没有人回答问题，我生气了！

`builtin_clz`：`builtin_clz`楼上

`builtin_clz`：`builtin_clz`楼下

`builtin_clz`：`builtin_clz`楼上

`builtin_clz`：`builtin_clz`楼下

……

虽然这个名叫 `builtin_clz` 的网友因为没有人回答他的学术问题被激怒了，但这个有趣的发言方式让小 I 乐呵了许久，这说明人类的悲欢并不相通。不过当小 I 刷新界面想要往下浏览大家的回复时，却发现学术社区的管理员因为这个帖子过于灌水把它删除了。

为了恢复这个有趣的帖子，小 I 对着网页缓存倒腾了许久，还原出了这个帖子的每条消息。然而因为神秘原因，消息的顺序被打乱了，且缓存中没有每条消息发送的时间，因而小 I 没有办法还原原始帖子中消息的顺序。

秉承“遇到困难睡大觉”精神的小 I 决定随便给帖子里的消息排个顺序，不过深受“XXX楼上”“XXX楼下”这种发言形式吸引的小 I 还是希望重排之后有尽可能多的这种形式的消息的表达是符合帖子的实际情况的。然而小 I 是一个只会水社区不会做题的 OI 选手，所以小 I 求助于你。

当然了，小 I 知道直接将帖子中的原始信息丢给你对你来说是不方便的，所以他对信息进行了一些规范化处理，详见题目描述中的形式化题意。**同时由于学术社区的特殊规定，帖子中的消息满足一定特殊限制，详见题目描述最后。**

【题目描述】

以下涉及的所有字符串判等操作都对大小写敏感，例如 loushang、Loushang、LOUSHANG 是互不相同的字符串。

小 I 正在整理学术社区中的一个帖子。帖子中共有 N 个网友发过消息，他们的网名分别为 $n_1, n_2, \cdots, n_N$。帖子中总共有 M 条消息，对于第 i 条消息，我们用三个字符串 $s_{i,1}, s_{i,2}, s_{i,3}$ 构成的三元组描述它，其中 $s_{i,1}$ 表示这条消息发出者的网名，而 $s_{i,2}$ 和 $s_{i,3}$ 描述这条消息的内容。

对于第 i 条消息，我们通过如下方式定义其属于**楼下型消息**、**楼上型消息**、**学术消息**中的哪一种：

- 若字符串 $s_{i,3}$ 为 louxia，且 $s_{i,2}$ 恰好与给出的某个网名相同（注意 $s_{i,2} = s_{i,1}$ 是允许的），则称这条消息是**楼下型消息**，$s_{i,2}$ 对应这条消息提到的网友；
- 若字符串 $s_{i,3}$ 为 loushang，且 $s_{i,2}$ 恰好与给出的某个网名相同（注意 $s_{i,2} = s_{i,1}$ 是允许的），则称这条消息是**楼上型消息**，$s_{i,2}$ 对应这条消息提到的网友；
- 若以上两个条件都不满足，则称这条消息是**学术消息**。

定义一个对所有消息的重排方案为一个 1 到 M 的排列 $a_1, a_2, a_3, \cdots, a_M$，表示第一条消息是 $(s_{a_1,1}, s_{a_1,2}, s_{a_1,3})$，第二条消息是 $(s_{a_2,1}, s_{a_2,2}, s_{a_2,3})$，依此类推。

对于一个重排方案 $a_1, a_2, \cdots, a_M$ 中的第 $i (1 \le i \le M)$ 条消息 $(s_{a_i,1}, s_{a_i,2}, s_{a_i,3})$，如下定义其是否是**符合实际情况的**：

- 若这条消息是**楼下型消息**，则这条消息是**符合实际情况的**当且仅当 $i \neq 1$ 且 $s_{a_{i-1},1} = s_{a_i,2}$，即上一条消息存在且它的发出者与这条消息提到的网友一致。
- 若这条消息是**楼上型消息**，则这条消息是**符合实际情况的**当且仅当 $i \neq M$ 且 $s_{a_{i+1},1} = s_{a_i,2}$，即下一条消息存在且它的发出者与这条消息提到的网友一致。
- 若这条消息是**学术消息**，则无论如何这条消息一定不是符合实际情况的，这是因为小 I 只想灌水不想学术。

在以上定义下，小 I 希望找到一个重排方案，使得该重排方案中符合实际情况的消息数量最多。你需要帮他找到这个方案以及这个方案中符合实际情况的消息数量。

为了方便你的解题，小 I 还告诉了你帖子中消息的一个特殊限制：因为学术社区会禁言在社区中只灌水不学术的人，所以在小 I 给出的帖子里，每一个在帖子中发过言的人都一定会在帖子中发出至少一条学术消息。

【输入格式】

从文件 ***community.in*** 中读入数据。

本题有多组测试数据。第一行一个整数 T 表示测试数据组数，接下来分别输入 T 组数据。

对于每组测试数据，第一行两个整数 N, M 分别表示帖子中发过消息的网友数量以及帖子的消息数量。

接下来 N 行每行一个字符串 n 表示在帖子中发过消息的一个网友的网名。保证每个测试数据中输入的 N 个网友的网名两两不同。

接下来 M 行每行三个字符串 s_1, s_2, s_3 描述一条消息，相邻的字符串之间用一个空格分隔，其中 s_1 一定与输入中某个网友的网名相等。

输入的所有字符串仅由大小写英文字母、下划线（_）、英文问号（?）、英文感叹号（!）和英文句号（.）构成，且长度不超过 12。

对于每组测试数据，保证输入的 N 个网名都发出过至少一条消息，**且至少发出过一条学术消息**。

同一组测试数据中可能存在多条消息内容完全一致，此时应将他们视为**多条**消息。

【输出格式】

输出到文件 ***community.out*** 中。

对于每组测试数据输出两行。

第一行输出一个非负整数表示所有重排方案中最大的符合实际情况的消息数量。

第二行输出 M 个整数 $a_1, a_2, \cdots, a_M$，表示符合实际情况的消息最多的重排方案。若有多种合法的重排方案，**输出任意一个即可**。

【样例 1 输入】

```
2
4 15
builtin_clz
builtin_ctz
jinkela
OrzTourist
builtin_clz MengXin QiuZhu
builtin_ctz builtin_clz louxia
jinkela builtin_ctz louxia
builtin_ctz builtin_clz loushang
builtin_clz BieMoZheng YaoXueShu
OrzTourist builtin_clz louxia
OrzTourist OrzTourist louxia
builtin_clz Iam Angry!
```

```
builtin_clz builtin_clz loushang
builtin_clz builtin_clz louxia
builtin_clz builtin_clz loushang
builtin_clz builtin_clz louxia
builtin_ctz Xue Shu
jinkela Xue Shu
OrzTourist Xue Shu
1 9
builtin_clz
builtin_clz builtin_clz loushang
builtin_clz builtin_clz loushang
builtin_clz builtin_clz louxia
builtin_clz builtin_clz Loushang
builtin_clz builtin_clz LOUSHANG
builtin_clz Builtin_clz loushang
builtin_clz loushang louxia
builtin_clz builtin_clz builtin_clz
builtin_clz loushang builtin_clz
```

【样例 1 输出】

```
9
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
3
8 1 2 7 9 3 6 4 5
```

【样例 1 解释】

第一个测试数据与题目背景中给出的例子基本一致，而不同的点在于：为了满足每个人至少发出一条学术消息的要求，在该组数据输入的最后有几条额外的学术消息。

第二个测试数据中，输入的前两条消息是楼上型消息，第三条消息是楼下型消息，其他消息是学术消息。

【样例 2】

见选手目录下的 ***community/community2.in*** 与 ***community/community2.ans***。

【样例 3】

见选手目录下的 ***community/community3.in*** 与 ***community/community3.ans***。
该组样例满足数据范围中的特殊性质 A、特殊性质 B、特殊性质 C。

【样例 4】

见选手目录下的 ***community/community4.in*** 与 ***community/community4.ans***。
该组样例满足数据范围中的特殊性质 C。

【数据范围】

设 $\sum M$ 为单个测试点中所有测试数据的 M 的和。
对于所有测试点，$1 \le T \le 10^2, 1 \le N \le M \le 77,777, 1 \le \sum M \le 2.5 \times 10^5$。

<table>
<tr><th>$T \le$</th><th>$M \le$</th><th>$\sum M \le$</th><th>测试点编号</th><th>特殊性质 A</th><th>特殊性质 B</th><th>特殊性质 C</th></tr>
<tr><td>5</td><td>10</td><td>50</td><td>1</td><td rowspan="2">无</td><td rowspan="2">无</td><td rowspan="2">无</td></tr>
<tr><td>10</td><td>16</td><td>160</td><td>2</td></tr>
<tr><td rowspan="5">30</td><td rowspan="5">2,222</td><td rowspan="5">15,000</td><td>3,4</td><td rowspan="2">有</td><td>有</td><td rowspan="3">有</td></tr>
<tr><td>5,6</td><td>无</td></tr>
<tr><td>7,8,9</td><td rowspan="3">无</td><td>有</td></tr>
<tr><td>10,11</td><td rowspan="2">无</td></tr>
<tr><td>12,13</td><td>无</td></tr>
<tr><td rowspan="5">10^2</td><td rowspan="5">77,777</td><td rowspan="5">2.5×10^5</td><td>14,15</td><td rowspan="2">有</td><td>有</td><td rowspan="3">有</td></tr>
<tr><td>16</td><td>无</td></tr>
<tr><td>17,18,19</td><td rowspan="3">无</td><td>有</td></tr>
<tr><td>20,21,22</td><td rowspan="2">无</td></tr>
<tr><td>23,24,25</td><td>无</td></tr>
</table>

注意：为了阅读方便，测试点编号在表格中的第四列。

特殊性质 A：没有楼上型消息。**注意：这不意味着 s_3 不等于 `loushang`。**

特殊性质 B：对于每组测试数据，存在一个重排方案，使得每一条楼上型消息和楼下型消息都是符合实际情况的。

特殊性质 C：对于每组测试数据，若存在一条消息是 s_1 s_2 `loushang`，其中 s_1, s_2 为任意字符串，则该组数据中一定不存在一条消息是 s_2 s_1 `louxia`。

【评分方式】

若一个测试点内所有测试数据的符合实际情况的消息数量都正确，你将获得该测试点 50% 的分数；在此基础上，若一个测试点内所有测试数据的重排方案都正确，你将

获得该测试点的所有分数。需要注意的是，**如果你只希望获得 50% 的分数，你也要保证在每组测试数据的第二行输出一个 1 到 M 的排列，否则实际分数与期望分数可能出现偏差**。

【提示】

因为这对你可能很重要，所以小 I 再一次强调：**因为学术社区会禁言在社区中只灌水不学术的人，所以在小 I 给出的帖子里，每一个在帖子中发过言的人都一定会在帖子中发出至少一条学术消息。**

本题输入规模较大，请使用较为快速的输入方式。