

2015 考研机试

对王道考研原文档题面与数据范围进行重构和补全。

T1 自强不息数

时间限制：1.0 秒

空间限制：512 MB

题目描述

若一个数的绝对值大于 k ，我们称之为“自强不息数”。给定 n 个数，求其中自强不息数的个数。

输入格式

第一行包含 2 个整数 n 和 k ($1 \leq n \leq 10^5$, $0 \leq k \leq 10^9$)。

接下来为 n 个用单个空格隔开的整数，每个数的绝对值不超过 10^9 。

输出格式

仅输出一个整数，表示自强不息数的个数

样例 #1

样例输入 #1

```
1 7 5
2 2 4 5 9 -9 -3 -7
```

样例输出 #1

```
1 3
```

T2 整数对

时间限制：1.0 秒

空间限制：512 MB

题目描述

已知 n 个非负数整数 A_1, A_2, \dots, A_n ，以及一个非负数整数 M 。请你输出满足以下条件的 (i, j) 的数对个数：

- $i < j$,
- $A_i + A_j \leq M$

由于满足条件的整数对可能很多，你只需要输出满足条件的整数对的个数除以 $10^9 + 7$ 的余数

输入格式

输入的第一行包含整数 n ($2 \leq n \leq 10^5$) 和 M ($0 \leq M \leq 10^9$), 以一个空格隔开

输入的第二行包含 n 个以一个空格分隔开的整数, 表示 A_1, A_2, \dots, A_n ($0 \leq A_i \leq 10^9$)

在 50% 的测试用例中, $2 \leq n \leq 1000$

输出格式

输出一个整数, 即满足条件的整数对的个数除以 $10^9 + 7$ 的余数

样例 #1

样例输入 #1

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

样例输出 #1

```
1 6
```

样例解释 #1

$(1, 2), (1, 3), (1, 4), (1, 5), (2, 3), (2, 4)$ 是所有满足条件的 (i, j) 组合。

T3 CSV 数据处理

时间限制: 1.0 秒

空间限制: 512 MB

题目描述

小明要处理一些以 CSV (Comma-Separated Values) 格式存储的数据文件。虽然并没有一个通行的 CSV 文件格式的标准, 通过观察, 小明对他所需要处理的 CSV 格式的数据总结如下:

一个 CSV 数据文件有若干行, 每行有若干个被逗号分隔开的项目。一个项目可以是以下几种数据中的一种:

- 一个字符串。本题中, 一个字符串是指 1 到 255 个连续出现的字符。字符串中可以出现的字符有大写字母, 小写字母, 空格, 逗号 , , 以及小数点 . 。
- 一个整数。本题中出现的整数一定在 $[-10^9, 10^9]$ 范围内。比如 43 , 12345678 , 0 都是合法的整数。
- 一个日期。在这种情况下, 日期一定会符合 $\langle \text{YEAR} \rangle / \langle \text{MONTH} \rangle / \langle \text{DAY} \rangle$ 的形式。如果月份或者日期是一位数, 则既可以写作一位数字也可以在前面补一个零。如 1980 年的 1 月 1 号可以被写作 1980/01/01 , 1980/1/1 , 1980/1/01 或者 1980/01/1 。本题中出现的日期中的年份一定在 1900 到 2100 年之间, 且日期合法。

不管是哪一种数据, 在 CSV 中出现时都有可能在两侧带上一对双引号, 处理的时候应该认为这个数据就是双引号之间的内容, 不含双引号。比如 "12" 表示一个整数 12 。"2015/3/14" 表示日期 2015 年 3 月 14 日, "CSV Co., Ltd." 是一个内容为 CSV Co., Ltd. 的字符串。双引号总是成对出现。

字符串中可能包含逗号，但是如果字符串包含逗号的话，则一定会用一对双引号括起来。此时的双引号内的逗号并不视为分隔两个项目的逗号。

此外，在数据两侧还可能有多余的空格，在处理的时候请忽略这些空格。但是如果空格在双引号内，则应当被看作字符串的一部分。

如果一对双引号里的内容不是一个字符串（也就是说，是一个整数或者一个日期），则该对双引号内不会出现空格。例如 "12 34" 表示一个字符串 12 34。而字符串不包含 / 字符，故只要双引号中的内容包含 /，则引号内的内容一定是一个合法的日期。

小明需要处理的 CSV 文件还满足以下性质：该文件的每行的项目个数恰好相同。第一行的所有项目都是字符串，且这些字符串只由**大写字母**组成，这些字符串是每一列的名字。列的名字**不会重复**。除了第一行以外的所有项目则是数据部分，这部分里任何一列的所有项目都属于同一类别（字符串，整数或日期）。你需要根据上述描述来判断每一列的内容所对应的数据类型。

小明希望你对这些数据进行以下两种类型的处理：

- 排序。命令格式为 `SORT <name>`。其中 `<name>` 为某一列的名字。你需要把所有数据根据指定列的内容进行排序：
 - 对于两个字符串 a 和 b ，如果 a 是 b 的前缀则 $a < b$ ，如果 b 是 a 的前缀则 $b < a$ 。否则， a 和 b 的大小关系由它们的第一个对应位置上不相同的字符决定。在这种情况下，在下表中越早出现的字符被认为越小，越应该排序排到前面（最左侧的字符是一个空格，为了阅读方便以一个下划线显示）：`_, .ABCDEFGHIJKLMN O PQRSTU VWXYZabcdefghijklmnopqrstuvwxyz`
 - 整数的排序方式是数值越小的数字排在越前面；
 - 日期的排序方式是日期越早排在越前面；
 - 不管是对哪一种数据进行排序，对于指定列内容相同的各行，都不应该改变它们的相对顺序。
- 筛选。命令格式为 `FILTER <name> <op> <value>`。其中 `<name>` 为某一列的名字，`<op>` 是比较运算符，可以是以下五种符号中的一种：`=, <, <=, >, >=`。你需要把所有数据根据指定列的内容进行筛选，保留指定列的值和 `<value>` 符合给定 `<op>` 关系的行，而删除剩余的行。例如，如果命令是 `FILTER ID > 5`，你就需要保留名字为 `ID` 的列的值大于 5 的所有行，而删除其余行。其余类型的比较关系和上述排序方式相同。保留的行之间的相对顺序不变。`<value>` 保证类型与指定列的数据类型相同。`<value>` 可能以双引号括起来。

在执行完给定的操作之后，小明希望你以之前描述过的 CSV 格式，把处理完的数据再输出出来。

小明希望你输出数据时不要在项目两侧添加任何多余的空格，并且只在**必须**用双引号的时候（即需要输出一个包含逗号或两侧有空格的字符串时）才使用双引号。此外，小明还要求你在输出日期的时候对于月份和日期一律输出两位数字（如果只有一位的话在前面补一个零）。

输入格式

从标准输入读入数据。

输入的第一行有两个正整数 n, m ，分别是待处理的 CSV 文件的长度和小明希望你进行操作的个数。

接下来的 n 行是需要你处理的 CSV 文件的内容，具体内容格式见题目描述一栏。

接下来的 m 行是小明希望你进行的操作，具体内容格式见题目描述一栏。你需要按顺序进行这些操作。

输出格式

输出到标准输出。

输出若干行（至少为 1 行），为执行完小明给定的操作以后的数据，按照题目描述一栏的格式进行输出。不难发现，行数最少的情况即为删除所有数据行，只保留表示每一列名字的第一行。

样例 #1

样例输入 #1

```
1 6 2
2 ID, NAME, BIRTHDATE, SCORE
3 1 , Xiao Ming , "1993/04/01" , 60
4 2, "Xiao Hong ",1993/05/05,98
5 3, "Zhang San" ,1993/10/27,33
6 4,Li Si,1994/12/25,"94"
7 5, "Wang Wu,,,,," ,1993/09/11,76
8 FILTER SCORE >= "60"
9 SORT BIRTHDATE
```

样例输出 #1

```
1 ID,NAME,BIRTHDATE,SCORE
2 1,Xiao Ming,1993/04/01,60
3 2,"Xiao Hong ",1993/05/05,98
4 5,"Wang Wu,,,,," ,1993/09/11,76
5 4,Li Si,1994/12/25,94
```

样例解释 #1

该样例所表示的 CSV 文件一共有 6 行，小明给出的操作有 2 次。

CSV 文件的第 1 行为各列的所有名字。接下来的第 2 ~ 6 行为具体信息。根据第 2 行的内容，可以判断出各列的数据类型：`ID` 为整数，`NAME` 为字符串，`BIRTHDATE` 为日期，`SCORE` 为整数，共计 4 列。

第一次操作，只保留 `SCORE` 大于等于 60 的行。故原始 CSV 文件的第 4 行被删除。

第二次操作，需要根据 `BIRTHDATE` 从小到大进行排序。

在输出 `NAME` 当中，`Xiao Ming` 中间的空格需要保留，右侧的空格需要去掉。`"Xiao Hong "` 内部最右侧的空格需要保留，输出时需要保留双引号。`"Wang Wu,,,,,"` 在输出时也需要保留双引号。`SCORE` 当中的 94 需要去掉引号再输出。若 CSV 文件的第 4 行未被删除，其 `NAME` 一栏的 `"Zhang San"` 需要去掉外层的双引号再输出。

子任务

对于所有数据，保证 $n \leq 1000$, $m \leq 10$ ，被处理的 CSV 文件的每行的项目不超过 10 个。

测试点编号	特殊性质
1 ~ 3	输入文件无双引号, 没有筛选操作
4 ~ 6	输入文件无双引号, 没有排序操作
7 ~ 8	输入文件无双引号
9 ~ 11	没有筛选操作
12 ~ 14	没有排序操作
15 ~ 20	无

T4 货物运输

时间限制： 1.0 秒

空间限制： 512 MB

题目描述

巨大的海域上分布着 n 个小岛，这些小岛由 m 座桥梁连接，每座桥梁连接两个小岛，所有桥梁都是双向通行的。这些桥梁能承受的重量都是有限的，不同桥梁承重可能不同。我们说一个桥梁的承重能力为 w ，如果一辆车可以运输重量为 w 的货物安全通过该桥梁，但是不能再多运一点点货物。

现在 R 公司要在海域上的某个小岛建立一个仓库，公司希望仓库的货物可以运往至少 k 个小岛（包括建立仓库的小岛）。求在上述条件下，每车货物的最大重量。

输入格式

从标准输入读入数据。

第一行包含三个整数 n, m, k 。

接下来 m 行，其中的第 i 行包括三个整数 u_i, v_i, w_i 。表示小岛 u_i 和小岛 v_i 之间有一座承重为 w_i 的桥梁。

输出格式

输出到标准输出。

输出一行，仅包括一个整数，表示每车货物的最大重量。若无论每车货物的重量为多少都无法使得货物可以运往至少 k 个小岛，则输出 0。

样例 #1

样例输入 #1

```
1 | 5 7 3
2 | 1 2 10
3 | 2 3 3
4 | 1 4 3
5 | 3 4 6
6 | 3 5 5
7 | 4 5 7
8 | 1 1 22
```

样例输出 #1

```
1 | 6
```

样例解释 #1

将仓库建在第 5 个小岛，重量为 6 的货物可以运往 3, 4, 5 这三个小岛。

数据范围

对于 40% 的数据，有 $2 \leq k \leq n \leq 10^3$, $1 \leq m \leq 5 \times 10^3$ 。

对于 100% 的数据，有 $2 \leq k \leq n \leq 10^5$, $1 \leq m \leq 5 \times 10^5$, $1 \leq w_i \leq 10^6$ 。

T5 新建道路

时间限制：1.0 秒

空间限制：512 MB

题目描述

C 国共有 n 个城市。有 $n - 1$ 条双向道路，每条道路连接两个城市，任意两个城市之间能互相到达。为了使得交通更加便捷，C 国政府打算新建一条连接两个城市的道路，使得某些城市之间的最短距离能够比原来更小。现在政府共提出了 m 个备选方案，你需要回答每个方案能够使得多少对城市的最短距离减小。

C 国的 1 号城市特别重要，因此，1 号城市到任意一个城市的简单路径上至多包括 100 个城市。

输入格式

第一行包含两个整数 n 和 m 。

接下来 $n - 1$ 行，每行包括三个整数 u, v, w 。表示城市 u 和城市 v 之间已经存在一条长度为 w 的道路。

接下来 m 行，每行包括三个整数 x, y, z 。描述了一个备选方案，该方案将在城市 x 和城市 y 之间建立一条长度为 z 的道路。需要注意的是，每一种备选方案都是在原始的道路系统上添加一条道路，各备选方案之间互不影响。

输出格式

对于每个方案，输出一行，仅包括一个整数，表示该方案能够使得多少对城市的最短距离减小。

样例 #1

样例输入 #1

```
1 4 3
2 1 2 2
3 2 3 4
4 2 4 5
5 1 3 1
6 1 4 6
7 2 2 1
```

样例输出 #1

```
1 3
2 1
3 0
```

样例解释 #1

原始道路系统当中，两两城市的最短路径如下：

- 1, 2 号城市的最短距离为 2；
- 1, 3 号城市的最短距离为 6；
- 1, 4 号城市的最短距离为 7；

- 2, 3 号城市的最短距离为 4 ;
- 2, 4 号城市的最短距离为 5 ;
- 3, 4 号城市的最短距离为 9 。

第一个备选方案在 (1, 3) 号城市添加一条长度为 1 的道路, 将 (1, 3), (2, 3), (3, 4) 的最短距离缩短, 故答案为 3。

第二个备选方案在 (1, 4) 号城市添加一条长度为 6 的道路, 将 (1, 4) 的最短距离缩短, 故答案为 1。

第三个备选方案在 2 号城市上添加了一条长度为 1 的自环, 没有缩短任何城市之间的最短距离, 故答案为 0。

数据范围

对于 10% 的数据, $n, m, q \leq 100$ 。

对于另外 20% 的数据, $n, m, q \leq 10^3$ 。

对于所有数据, 保证 $n, m, q \leq 10^5$, 保证原始道路系统一定构成一棵树, 且所有城市到 1 号城市的简单路径上至多包括 100 个城市。 $1 \leq u, v, x, y \leq n, 0 \leq w, z \leq 10^5$ 。