

非确定机

【问题描述】

“非确定机”是现在假想中的一种计算机，它可以同时运行任意多段指令。这种计算机中允许一种新的分支指令，执行到这条指令时，程序会一分为二，同时分别执行这两个分支。

“非确定机”的一个神奇的功能是程序反转。给定一个程序和该程序的输出，它可以用相同的时空代价得到一个符合该输出的输入。这道题目正是要你反转运行一个程序。

在本题中，你有一个已编译好的，包含一些算法的程序 *prog*。它的输入为一个有向图 G 和算法编号 k ，输出为在有向图 G 上运行算法 k 得到的结果。同时，你会得到 10 个由程序 *prog* 运行得到的输出文件。你的任务是对每个输出文件给出一个可能的输入文件。

为了区分，在后文中如果没有特殊说明，我们把给定的程序 *prog* 的输出文件称作“输入”，把你需要提交的程序 *prog* 的输入文件称作“输出”。

【输入格式】

该题为提交答案型试题，所有输入数据 *nm1.in~nm10.in* 已在试题目录下。

输入的第一行包含三个正整数 n, k, p ，表示图 G 的点数，当前使用算法和一个由图 G 计算出的评分参数。

接下来有若干行，每行包含若干个整数，其意义需要你去探究。

【输出格式】

针对给定的 10 个输入文件 *nm1.in~nm10.in*，你需要分别提交你的输出文件 *nm1.out~nm10.out*。

输出文件的第一行包含 3 个整数 n, m, k ，表示有向图 G 的点数，边数和当前使用的算法。

接下来有 m 行，每行包含 3 个整数 u, v, w ，表示一条从节点 u 连向节点 v ，权值为 w 的有向边。其中节点的编号用 1 到 n 的整数表示。要求 w 必须为不超过 20000 的非负整数，且不能出现重边。允许出现自环。

【程序的使用】

在终端中先切换到该试题的目录下

```
cd nm
```

程序 *prog* 已在试题目录下，其使用方法是

```
./prog <output_file>
```

程序会把 *<output_file>* 作为程序 *prog* 的输入，将运行的结果输出到标

准输出。例如

`./prog nm4.out`

程序将会把 `nm4.out` 作为程序 *prog* 的输入。

如果你的文件不合法，程序将会输出错误信息。

【样例输入】

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

【样例输出】

```
3 3 0
1 3 92
2 1 12
3 2 29
```

【样例说明】

在样例中，边的权值 w 并不会影响到运行结果，在实际的数据中也要注意可能存在这种情况。

【评分方法】

每个测试点单独评分。

如果你的输出运行后得到的 n, k 与输入文件一致，得 1 分。

如果你的输出运行后得到的 n, k, p 与输入文件一致，得 2 分。

如果你的输出运行后得到的整数中，除了 p 以外均与输入文件一致，得 4 分。

在评测时，每个测试点会有一个评分参数 s 。如果你的输出运行后得到的整数中，除了 p 以外均与输入文件一致，且 p 值与标准文件差值的绝对值不超过 s ，得 7 分。

如果你的输出运行后得到的整数和输入文件全部一致，得 10 分。

以上条件如果满足多个，取最高分。

每个测试点的 s 如下

测试点编号	s	测试点编号	s
1	0	6	8
2	1000	7	20
3	0	8	1000

4	1000	9	1000
5	0	10	0

【如何测试你的输出】

程序 `prog` 还有测试的功能，可以根据你的输入输出文件以及 s 值给出得分或错误信息。具体用法为

```
./prog <output_file> <input_file> <s>
```

例如要测试测试点 6，可以使用

```
./prog nm6.out nm6.in 8
```

【提示】

题目中大多数 k 的值由两位数构成，若两种算法 k 的第一位相同，表示这两种算法采用了相同的基本算法。

请妥善保存输入文件 `nm*.in` 和你的输出 `nm*.out`，及时备份，以免误删。