Day 1

|  |  |  |  |
| --- | --- | --- | --- |
| 题目名称 | 动态图联通块 | 方城 | 连连看 |
| 可执行文件名 | lct | city | flow |
| 输入文件名 | lct.in | city.in | flow.in |
| 输出文件名 | lct.out | city.out | flow.out |
| 每个测试点时限 | 1秒 | 1秒 | 1秒 |
| 内存限制 | 256MB | 256MB | 256MB |
| 是否有部分分 | 否 | 否 | 否 |
| 题目类型 | 传统型 | 传统型 | 提交答案型 |
| 是否有样例文件 | 否 | 否 | 否 |
| 是否有附加文件 | 否 | 否 | 否 |

提交源程序须加后缀

|  |  |  |  |
| --- | --- | --- | --- |
| 对于C++ 语言 | lct.cpp | city.cpp | flow.cpp |

编译开关

|  |  |  |  |
| --- | --- | --- | --- |
| 对于C++ 语言 | -O2 -lm | -O2 -lm | -O2 -lm |

nodesnodesnodes1.in~nodes10.innodes1.out~nodes10.outnodes1.ans~nodes10.ans

动态图联通块(Lct)

【问题描述】

给定一棵个节点的树和次询问：编号在之间的节点形成的联通块的数量。

【输入格式】

输入文件为***lct.in***。

第一行包含2个正整数。

接下来行,每行包含2个正整数，表示一条无向边连接节点和节点。

接下在行,每行包含2个正整数。

【输出格式与部分分】

输出文件为***lct .out***。

共行包含1个整数，表示形成的联通块的数量。

【样例1输入】

4 6

1 4

4 3

3 2

1 2

2 3

3 4

1 3

2 4

1 4

【样例1输出】

2

1

1

2

1

1

【子任务】

对于的数据，

对于的数据，

对于的数据，

对于的数据，

对于的数据，

方城(City)

【问题描述】

对于一个大小为行列的自然数矩阵，定义 为矩阵第行的最大值， 为矩阵列的最大值。

现在给定A 和B，询问满足条件的矩阵的数量，对22222223 取模。如果无解则输出No solution!。

【输入格式】

输入文件为***city.in***。

第一行包含个正整数。

第二行包含个整数。

第三行包含个整数。

【输出格式与部分分】

输出文件为***city.out***。

共一行包含1 个整数或者’No solution!’。

【样例1输入】

2 2

1 1

1 1

【样例1输出】

7

【样例2输入】

4 3

2 3 4 3

1 3 4

【样例2输出】

12675

【样例3输入】

5 2

50 50 50 50 100

100 100

【样例3输出】

15171509

【子任务】

所有测试数据的范围和特点如下:

对于的数据，

对于的数据，

对于的数据，

连连看(Flow)

作为提交答案题，你并不需要提交flow.cpp。取而代之的，你需要提交flow1.out到flow10.out。

【问题背景】

最近小T迷上了一款连连看游戏，这个游戏是在一个的棋盘上进行的。上面一共有对不同颜色的点。游戏的目标是用线将相同颜色的点对连起来，并且用线将整个棋盘空白的格子全部填满。注意如果仅仅将点对全部连接，但是却没有填满整个棋盘，游戏依然是失败的。

现在小游戏一共有10个关卡，分别为flow1.in 到flow10.in，小T 希望你能够帮助他通过尽量多的关卡。当然既然是游戏，每一个关卡都是必然有解的。

【输入格式】

输入文件为flow.in。

第一行包含2个整数。

接下来行每行包含个字符。每个字符可以为一个点‘.’表示空格，或者前个字母中的一个，表示一种特定的颜色。每一个字母只会至多出现一次，同一个字母的大小写对应相同的颜色。

【输出格式】

输出文件为flow***.out***。

前行每行包含个字符，每个字符都是前个字母中的一个。其中小写字母至多一个，大写字母可能有多个。相同的字母对应相同的颜色，用来描述填满之后的棋盘。

接下来c行，每行的格式为‘’。其中 为第i 个大写字母， 为一个字符串，描述一条，由在输入数据中大写字母所在格子连向小写字母所在格子的路径。 由‘L’（向左），‘R’（向右），‘U’（向上）和‘D’（向下）四种字符组成。注意 的长度至少为1，最长不超过。

【样例】

查看下发的flow0.in/flow0.out

【附加文件】

下发在选手目录下的checker.cpp 可以自行编译来检验选手输出文件的合法性。使用方法为：在命令行输入: ./checker < input > < output >。