

# 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
-----------	---------	---------	---------

## 动态图联通块(Lct)

### 【问题描述】

给定一棵 $n$ 个节点的树和 $q$ 次询问：编号在 $[l_i, r_i]$ 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 $n, q$ 。

接下来 $n - 1$ 行,每行包含 2 个正整数 $x_i, y_i$ ，表示一条无向边连接节点 $x_i$ 和节点 $y_i$ 。

接下在 $q$ 行,每行包含 2 个正整数 $l_i, r_i$ 。

### 【输出格式与部分分】

输出文件为 *lct.out*。

共 $q$ 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10% 的数据,  $1 \leq n, q \leq 10^1$

对于20% 的数据,  $1 \leq n, q \leq 10^2$

对于30% 的数据,  $1 \leq n, q \leq 10^3$

对于40% 的数据,  $1 \leq n, q \leq 10^5$

对于100% 的数据,  $1 \leq n, q \leq 2 \times 10^5$

## 方城(City)

### 【问题描述】

对于一个大小为 $n$ 行 $m$ 列的自然数矩阵，定义 $A_i$  为矩阵第 $i$ 行的最大值， $B_i$  为矩阵 $j$ 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2 个正整数 $n, m$ 。

第二行包含 $n$ 个整数 $A_1, A_2, \dots, A_n$ 。

第三行包含 $m$ 个整数 $B_1, B_2, \dots, B_m$ 。

### 【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 10 % 的数据， $1 \leq n \leq 8$ ， $1 \leq m \leq 8$

对于 40 % 的数据， $1 \leq n \leq 8$ ， $1 \leq m \leq 100$

对于 100 % 的数据， $1 \leq n \leq 100$ ， $1 \leq m \leq 100$ ， $0 \leq A_i, B_i \leq 10^4$

## 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  $n \times n$  的棋盘上进行的。上面一共有  $c$  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  $n, c$ 。

接下来  $n$  行每行包含  $n$  个字符。每个字符可以为一个点 ‘.’ 表示空格，或者前  $c$  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  $n$  行每行包含  $n$  个字符，每个字符都是前  $c$  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 ‘ $C_i : Path_i$ ’。其中  $C_i$  为第  $i$  个大写字母， $Path_i$  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。 $Path_i$  由 ‘L’（向左），‘R’（向右），‘U’（向上）和 ‘D’（向下）四种字符组成。注意  $Path_i$  的长度至少为 1，最长不超过  $n \times n$ 。

### 【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>`。