

NOI2015山东省队选拔赛第二轮

第二天 (day2)

竞赛时间：2015年5月17日上午8:00-13:00

题目名称	模拟电路	嫁接树	立体图
英文题目名称	circuit	graft	cube
输入文件名	circuit.in	graft.in	cube.in
输出文件名	circuit.out	graft.out	cube.out
提交文件名	circuit.pas/c/cpp	graft.pas/c/cpp	cube.pas/c/cpp
试题类型	传统型	传统型	传统型
时间限制	1秒	2秒	1秒
空间限制	512M	512M	512M
测试点数目	20	20	20
是否有部分分	无	无	无

1 模拟电路(circuit)

一家著名的芯片公司希望您能帮他们在一些最新的产品上安装可以稳定电压的组件。每块芯片都被设计成 $N \times N$ 的带插槽的正方形，一个插槽可以安装一块单独的组件，你的任务是尽可能多地插入这些组件。

现代处理器的设计是很复杂的。为了可以稳定电压，你要面对下面几个限制：

- 一些插槽是不可用的。
- 一些插槽已经被其它的组件占据了，因而无法被新的组件使用。
- 内存总线要连接到芯片的水平和垂直的边界上，它们的负载电压需要是安全的。具体来说，芯片公司提供了 N 组限制条件，分别对应了 N 行。其中对于第 i 行以及第 i 组限制条件，给定非负整数 T_i 与 T_i 个列编号，记为 r_{ij} ($1 \leq j \leq T_i$)，要求满足：第 i 行的组件数目不能超过指定的 T_i 个列方向上组件数目的和（也就是不超过“第 r_{i1} 列的组件数目+第 r_{i2} 列的组件数目+...”）。
- 为了避免插槽过热，给定浮点数 s_i ($1 \leq i \leq N$) 且 $0 \leq s_i \leq 1$ ，要求第 i 行的组件数不超过总组件数的 s_i 。同样给定浮点数 t_i ($1 \leq i \leq N$) 且 $0 \leq t_i \leq 1$ ，要求第 i 列的组件数不超过总组件数的 t_i 。

需要注意的是，已经占据了位置的组件，在统计一行或一列组件总数时，也是要被考虑在内的。而在计算芯片总组件数时，也要将已经占据了位置的组件考虑进去。

芯片被描述为一个 N 行，每行 N 个字符的矩阵，其中'.'表示开放插槽，'/'表示不可用插槽，'C'表示插槽已被一个组件占据。

输入：

第一行给定1个正整数，表示芯片的规模 N ($1 \leq N \leq 40$)。

然后给出 N 行，每行 N 个字符描述插槽，字符为'.'，'/'或'C'之一，含义如上所述。

之后 N 行，描述了限制条件。其中第 i 行首先给出非负整数 T_i ，表示第 i 行的限制条件涉及到的列的个数。之后再给出 T_i 个不重复的整数（都在1到 N 的范围中），描述了相关的列。

再下一行，给出了 N 个浮点数依次对应 s_i ， ($1 \leq i \leq N$)。每一个数字小数点后不超过三位。

再下一行，给出了 N 个浮点数依次对应 t_i ， ($1 \leq i \leq N$)。每一个数字小数点后不超过三位。

输出：

如果存在合法的策略，输出最多可以再在芯片上安装多少个组件。否则输出“impossible”（不用输出分号）。

数据规模：

对于5%的数据， $1 \leq N \leq 4$ 。

存在45%的数据， T_i 恒为0。其中更有15%的数据，不存在已经占据了位置的组件。

存在20%的数据， T_i 恒为1，且一定有 $r_{i1} = i$ 。

对于100%的数据， $1 \leq N \leq 40$ 。

样例：

- 样例输入一：

```
5
CC/..
./././
..C.C
/.C..
/./C/
1 1
1 2
1 3
1 4
1 5
0.3 0.3 0.3 0.3 0.3
0.3 0.3 0.3 0.3 0.3
```

- 样例输出一：

```
7
```

- 样例输入二：

```
5
CC/..
./././
..C.C
/.C..
/./C/
1 1
1 2
1 3
1 4
1 5
0.2 0.2 0.2 0.2 0.2
0.2 0.2 0.2 0.2 0.2
```

- 样例输出二：

```
impossible
```

样例说明：

对于样例一来说，每行每列上的组件不可以超过组件总数的 $\frac{3}{10}$ ，那么这块 5×5 的芯片的可被安装的最大组件数是7。

下图给出的是一种可行方案，其中'W'表示新加到开放插槽中的组件。

```
CC/W.
W/W//
W.C.C
/.CWW
/W/C/
```

2 嫁接树(graft)

Alice设计了一个树结构，有 N 个结点（包括根）被依次编号为1到 N ，由 $N - 1$ 条边连接。后来，Bob在上面增加了 K 条原来没有的边（也就是说既不是自环，也不会因此产生重边）并称这样得到的图为“ K -嫁接树”。

现在Alice希望对嫁接树的每一个结点进行染色，允许使用的颜色恰有 N 种，分别编号为1到 N 。Alice要求相邻两个结点要涂上不同的颜色。

假设颜色为 i 的结点有 t_i 个，则Bob给出了如下的评价分数：

$$score = \frac{t_1 + \frac{t_2}{2} + \frac{t_3}{3} + \cdots + \frac{t_N}{N}}{1 + P(t_1 + 2t_2 + 3t_3 + \cdots + Nt_N)}$$

其中 P 为非负系数。现在，Alice希望可以找到一种染色方案，使得Bob给出来的评分最大。你能帮助他吗？

输入：

第一行有2个整数，依次为 N 和 K ，如题所述。

第二行到第 $N + K$ 行，每行有两个整数 u 和 v ，依次给出了 $N + K - 1$ 条边。保证不存在自环，也不存在重边。

最后一行给定非负浮点数 P 。

输出：

输出最大的可能评分，四舍五入保留到小数点后第三位。

数据规模：

对于50%的数据， $K = 0$ ， $1 \leq N \leq 100000$ ， $0 \leq P < 10$ 。其中有25%的数据 $P = 0$ 。

对于另外50%的数据， $K \leq 2$ ， $1 \leq N \leq 20000$ ， $0 \leq P < 10$ 。其中也有25%的数据 $P = 0$ 。

样例数据：

● 样例输入一：

```
9 0
1 2
1 3
1 4
1 5
2 6
2 7
2 8
2 9
2.5
```

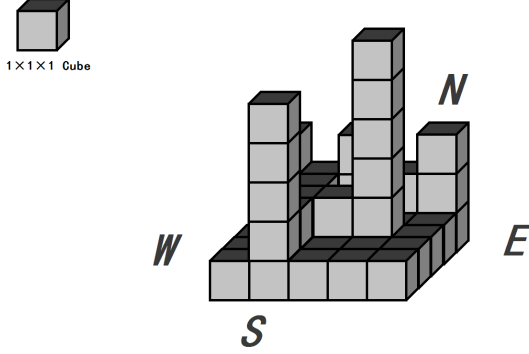
● 样例输出一：

```
0.253
```

3 立体图(cube)

小渊是个聪明的孩子，他经常会给周围的小朋友们讲些自己认为有趣的内容。最近，他准备给小朋友们讲解彩色水平光源照射下的立体图，并请你帮他在平面上画出来。

小渊有一块面积为 $m * n$ 的矩形区域，上面有 $m * n$ 个边长为1的格子，每个格子上堆了一些同样大小的积木（积木的长宽高都是1）。为了方便阐述，我们假设这块区域是坐北朝南的，下面我们给出一个例子。



小渊想请你打印出这些格子的立体图。我们定义每个积木为如下格式，并且不会做任何翻转旋转，只会严格以这一种形式摆放（左侧是应该打印出来的图样，右侧为对应每一个位置符号的十进制ASCII，其中ASCII为32的符号为空格）：

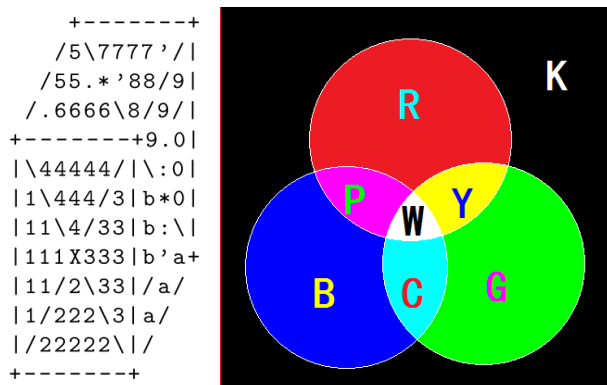
+-----+	[32, 32, 32, 32, 43, 45, 45, 45, 45, 45, 45, 45, 43]
/ \ ' /	[32, 32, 32, 47, 32, 92, 32, 32, 32, 32, 39, 47, 124]
/ . * ' /	[32, 32, 47, 32, 32, 46, 42, 39, 32, 32, 47, 32, 124]
/ . \ / /	[32, 47, 46, 32, 32, 32, 32, 92, 32, 47, 32, 47, 124]
+-----+ .	[43, 45, 45, 45, 45, 45, 45, 45, 43, 32, 46, 32, 124]
\ / \ :	[124, 92, 32, 32, 32, 32, 32, 47, 124, 92, 58, 32, 124]
\ / *	[124, 32, 92, 32, 32, 32, 47, 32, 124, 32, 42, 32, 124]
\ / : \	[124, 32, 32, 92, 32, 47, 32, 32, 124, 32, 58, 92, 124]
X ' +	[124, 32, 32, 32, 88, 32, 32, 32, 124, 32, 39, 32, 43]
/ \ / /	[124, 32, 32, 47, 32, 92, 32, 32, 124, 47, 32, 47, 32]
/ \ /	[124, 32, 47, 32, 32, 32, 92, 32, 124, 32, 47, 32, 32]
/ \ /	[124, 47, 32, 32, 32, 32, 92, 32, 124, 47, 32, 32, 32]
+-----+	[43, 45, 45, 45, 45, 45, 45, 45, 43, 32, 32, 32, 32]

在良好的光学试验环境下，小渊将 T 束平行光同时照射在这些积木上。这些平行光首先满足一定是红绿蓝三基色之一，其次入射角度满足：与 x 轴 y 轴的夹角度数均为45的倍数；且与 z 轴正方向的夹角或为45度，或为0度，或为315度。具体来说，我们最多会考虑9个方向的不同平行光，它们的入射方向可以被描述为：

- 西北方45度仰角 正北方45度仰角 东北方45度仰角
- 正西方45度仰角 垂直从上入射光 正东方45度仰角
- 西南方45度仰角 正南方45度仰角 东南方45度仰角

对于每一个单位积木来说，可以打印出来的三个表面被分为12个小三角形，如下图所示（下图中用符号0,1,2,...,9,a,b 分别表示了12个小三角形的区域）。

可以证明，每一个小三角形上照射到的光总是相同的。如果一个面上受到了多重基色光的照射，将得到叠加的颜色，它们的叠加方式如下图所示。同时需要注意的是，同色光之间不能产生能量的叠加：也就是说两束红色光与一束绿色光叠加成的依然是黄色光；而一束红色光与两束绿色光叠加成的也是黄色光，这两种叠加出来的黄色光是没有差异的。



- 红绿蓝三基色分别用字母 R G B 来表示。
- 而二次叠加后的三种颜色黄青紫，分别用 Y C P 来表示。
- 对于三次叠加后的颜色，也就是白色，用 W 来表示。
- 对于没有任何光线照射到的区域，呈现黑色，用 K 来表示。

输入格式：

第一行有两个正整数，分别是 m 和 n ，表示区域有 m 行，有 n 列。

之后 m 行，依次由远及近描述了每一行的情况。每一行给出 n 个正整数，表示第 i 行第 j 列中有堆放了多少积木。

之后 3 行，每行三个字符，描述了 9 个对应方向(与地图描述方向相同)的光照颜色。其中每一个字符或者为 R G B 中之一，表示对应的三种基色。或者为 $*$ ，表示没有照射光。

这九个方向依次是：

西北方 45 度仰角，正北方 45 度仰角，东北方 45 度仰角。

正西方 45 度仰角，垂直从上照射光，正东方 45 度仰角。

西南方 45 度仰角，正南方 45 度仰角，东南方 45 度仰角。

输出格式：

输出给出了打印后的效果。

其中要求输出结果不含前导空行，结尾也没有额外空行。输出的第一列不能全是空格，且每一行末尾也没有额外的空格。

数据规模：

对于 15% 的数据， $1 \leq m \leq 100$ 且 $1 \leq n \leq 100$ ，没有入射光。

对于 40% 的数据， $1 \leq m \leq 100$ 且 $1 \leq n \leq 100$ ，入射光只有一束，且入射方向一定是东南方。

对于 100% 的数据， $1 \leq m \leq 100$ 且 $1 \leq n \leq 100$ ，每一个位置堆放的积木总数不超过 100，入射光颜色可能是 R 或 G 或 B 中的任何一种颜色，最多可以有 9 束入射光。

此外，存在 25% 的数据，总的积木个数不超过 1000，也就是说建筑物的总体积不超过 1000。

样例数据：

- 样例输入一：

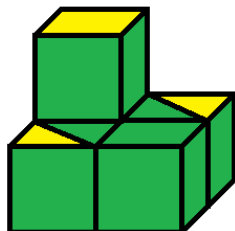
```
2 2
2 1
1 1
R**
***
**G
```

- 样例输出一：

```

      +-----+
      /Y\YYYY ' / |
      /YY.*'YY/G |
      /.YYYY\Y/G/ |
+-----+G.G |
| \GGGGG/ | \ : G |
| G\GGG/G | G * G |
| GG\G/GG | G : \ |
| GGGXGGG | G ' G +-----+
| GG/G\GG | /G/G\YYYY ' / |
| G\GGG\G | G/GG.*'YY/G |
| /GGGGG\ | / .GGGG\Y/G/ |
+-----+-----+G.G |
      /Y\GGGG ' /G\GGGG ' / | \ : G |
      /YY.*'GG/GG.*'GG/G | G * G |
      /.YYYY\G/.GGGG\G/G/ | G : \ |
+-----+-----+G.G | G ' G +
| \GGGGG/ | \GGGGG/ | \ : G | /G/
| G\GGG/G | G\GGG/G | G * G | G/
| GG\G/GG | GG\G/GG | G : \ | /
| GGGXGGG | GGGXGGG | G ' G +
| GG/G\GG | GG/G\GG | /G/
| G\GGG\G | G\GGG\G | G/
| /GGGGG\ | /GGGGG\ | /
+-----+-----+
```

- 样例一图示：



● 样例输入二：

```
3 4
1 1 2 1
1 2 1 2
2 1 2 1
**B
***
R*G
```

● 样例输出二：

```

+-----+
/W\WWWW' /|
/WW.*'WW/C|
/.WWWW\W/C/|
+-----+-----+-----+
/W\WWWW' /|\YYYYY/W\WWWW' /|
/WW.*'WW/C|G\YYY/WW.*'WW/C|
/.WWWW\W/C/|GG\Y/.WWWW\W/C/|
+-----+-----+-----+C.C|---+
/W\WWWW' /|\YYYYY/W\WWWW' /|\YYYYY/|\:C|C' /|
/WW.*'WW/C|G\YYY/WW.*'WW/C|G\YYY/Y|C*C|C/C|
/.WWWW\W/C/|GG\Y/.WWWW\W/C/|GG\Y/Y|C:\|/C/|
+-----+C.G|GGG+-----+C.G|GGGXYYY|C'C+C.C|
|\YYYYY/|\:G|GG/|\YYYYY/|\:G|GG/G\YY|/C/|\:C|
|Y\YYY/Y|C*G|G/K|Y\YYY/Y|C*G|G/GGG\Y|C/C|C*C|
|YY\Y/Y|C:\|/KK|YY\Y/Y|C:\|/GGGGG\|/C/|C:\|
|YYYYXYYY|C'G+---|YYYYXYYY|C'G+-----+C.C|C'C+
|YY/Y\YY|/G/G\KK|YY/Y\YY|/G/G\GGGG' /|\:C|/C/
|Y/YYY\Y|G/GG.*'|Y/YYY\Y|G/GG.*'WW/C|C*C|C/
|/YYYYY\|/.YYYYY\|/YYYYY\|/.WWWW\W/C/|C:\|/
+-----+-----+-----+C.C|C'C+
|\YYYYY/|\YYYYY/|\YYYYY/|\YYYYY/|\:C|/C/
|Y\YYY/Y|Y\YYY/Y|Y\YYY/Y|Y\YYY/Y|C*C|C/
|YY\Y/Y|YY\Y/Y|YY\Y/Y|YY\Y/Y|C:\|/
|YYYYXYYY|YYYYXYYY|YYYYXYYY|YYYYXYYY|C'C+
|YY/Y\YY|YY/Y\YY|YY/Y\YY|YY/Y\YY|/C/
|Y/YYY\Y|Y/YYY\Y|Y/YYY\Y|Y/YYY\Y|C/
|/YYYYY\|/YYYYY\|/YYYYY\|/YYYYY\|/
+-----+-----+-----+-----+
```

● 样例二图示：

