HDU1043

题目描述（HDU1043）：十五数码问题是由 15 块滑动的方块构成的，在每一块上有一个 1～15 的数字，所有方块都是一个 4×4 的排列，其中一块方块丢失，称之为“x”。拼图的目的是排列方块，使其按以下顺序排列：

```
1   2   3   4
5   6   7   8
9   10  11  12
13  14  15  x
```

其中唯一合法的操作是将“x”与相邻的方块之一交换。下面的移动序列解决了一个稍微混乱的拼图：

```
1  2  3  4      1  2  3  4      1  2  3  4      1  2  3  4
5  6  7  8      5  6  7  8      5  6  7  8      5  6  7  8
9  x  10 12     9  10 x  12     9  10 11 12     9  10 11 12
13 14 11 15     13 14 11 15     13 14 x  15     13 14 15 x
            r->             d->             r->
```

上一行中的字母表示在每个步骤中“x”方块的哪个邻居与“x”交换；合法值分别为“r”“l”“u”和“d”，表示右、左、上和下。

在这个问题中，编写一个程序来解决八数码问题，它由 3×3 的排列组成。

输入：输入包含多个测试用例，描述是初始位置的方块列表，从上到下列出行，在一行中从左到右列出方块，其中的方块由数字 1～8 加上“x”表示。例如以下拼图:

```
1 2 3
x 4 6
7 5 8
```

由以下列表描述：

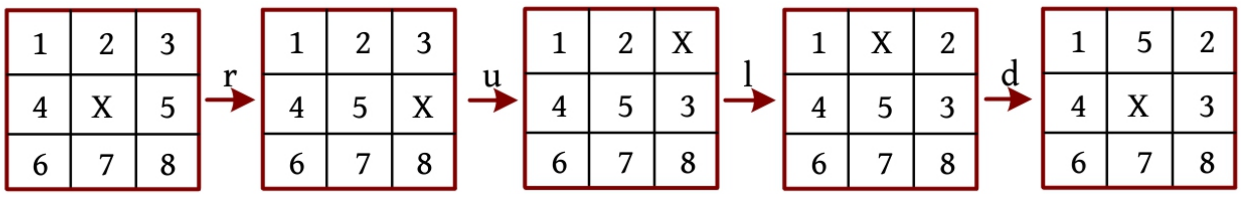
```
1 2 3 x 4 6 7 5 8
```

输出：如果没有答案，则输出“unsolvable”，否则输出由字母“r”“l”“u”和“d”组成的字符串，描述产生答案的一系列移动。字符串不应包含空格，并从行首开始。

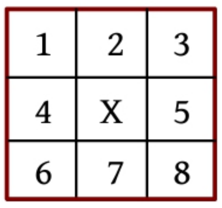
输入样例	输出样例
2 3 4 1 5 x 7 6 8	ullddrurdllurdruld

HDU3567

描述（HDU3567）：八数码，也叫作“九宫格”，来自一个古老的游戏。在这个游戏中，你将得到一个 3×3 的棋盘和 8 个方块。方块的编号为 1～8，其中一块方块丢失，称之为“X”。“X”可与相邻的方块交换位置。用符号“r”表示将“X”与其右侧的方块进行交换，用“l”表示左侧的方块，用“u”表示其上方的方块，用“d”表示其下方的方块。



棋盘的状态可以用字符串 S 表示，使用下面显示的规则。



用“1234X5678”表示

问题是使用“r”“u”“l”“d”操作列表可以将棋盘的状态从状态 A 转到状态 B，需要找到满足以下约束的结果：

（1）在所有可能的解决方案中，它的长度最小；

（2）它是所有最小长度解中词典序最小的一个。

输入：第 1 行是 T（T≤200），表示测试用例数。每个测试用例的输入都由两行组成，状态 A 位于第 1 行，状态 B 位于第 2 行。保证从状态 A 到状态 B 都有有效的解决方案。

输出：对于每个测试用例，都输出两行。第 1 行是“Case x:d”格式，其中 x 是从 1 开始计算的案例号，d 是将 A 转换到 B 的操作列表的最小长度。第 2 行是满足约束条件的操作列表。

输入样例	输出样例
2	Case 1: 2
12X453786	dd
12345678X	Case 2: 8
564178X23	urrulldr
7568X4123	

POJ2449

题目描述（POJ2449）：给定一个有向图，N 个节点，M 条边。求从源点 S 到终点 T 的第 K 短路。路径可能包含两次或两次以上的同一节点，甚至是 S 或 T。具有相同长度的不同路径将被视为不同。

输入：第 1 行包含两个整数 N 和 M（1≤N≤1000，0≤M≤100 000）。节点编号为 1～N。以下 M 行中的每一行都包含 3 个整数 A、B 和 T（1≤A, B≤N，1≤T≤100），表示从 A 到 B 有一条直达的路径，需要时间 T。最后一行包含 3 个整数 S、T 和 K（1≤S, T≤N，1≤K≤1000）。

输出：单行输出第 K 短路径的长度（所需时间）。如果不存在第 K 短路，则输出-1。

输入样例	输出样例
2 2 1 2 5 2 1 4 1 2 2	14

POJ3134

题目描述（POJ3134）：从 x 开始，反复乘以 x，可以用 30 次乘法计算 x^{31}：$x^2=x\times x$，$x^3=x^2\times x$，$x^4=x^3\times x$，…，$x^{31}=x^{30}\times x$。

平方运算可以明显地缩短乘法序列，以下是用 8 次乘法计算 x^{31} 的方法：$x^2=x\times x$，$x^3=x^2\times x$，$x^6=x^3\times x^3$，$x^7=x^6\times x$，$x^{14}=x^7\times x^7$，$x^{15}=x^{14}\times x$，$x^{30}=x^{15}\times x^{15}$，$x^{31}=x^{30}\times x$。

这不是计算 x^{31} 的最短乘法序列。有很多方法只有 7 次乘法，以下是其中之一：$x^2=x\times x$，$x^4=x^2\times x^2$，$x^8=x^4\times x^4$，$x^{10}=x^8\times x^2$，$x^{20}=x^{10}\times x^{10}$，$x^{30}=x^{20}\times x^{10}$，$x^{31}=x^{30}\times x$。

如果除法也可用，则可以找到一个更短的操作序列。可以用 6 个运算（5 乘 1 除）计算 x^{31}：$x^2=x\times x$，$x^4=x^2\times x^2$，$x^8=x^4\times x^4$，$x^{16}=x^8\times x^8$，$x^{32}=x^{16}\times x^{16}$，$x^{31}=x^{32}\div x$。

如果除法和乘法一样快，则这是计算 x^{31} 最有效的方法之一。

编写一个程序，通过从 x 开始的乘法和除法，为给定的正整数 n 找到计算 x^n 的最少运算次数。在序列中出现的乘积和商应该是 x 的正整数幂。

输入：输入是由一行或多行组成的序列，每行都包含一个整数 n（0<n≤1000）。以输入 0 结束。

输出：单行输出从 x 开始计算 x^n 所需的最小乘法和除法总数。

输入样例	输出样例
1 31 70 91 473 512 811 953 0	0 6 8 9 11 9 13 12