**Describe：**

The game “The Pilots Brothers: following the stripy elephant” has a quest where a player needs to open a refrigerator.

There are 16 handles on the refrigerator door. Every handle can be in one of two states: open or closed. The refrigerator is open only when all handles are open. The handles are represented as a matrix 4х4. You can change the state of a handle in any location [i, j] (1 ≤ i, j ≤ 4). However, this also changes states of all handles in row i and all handles in column j.

The task is to determine the minimum number of handle switching necessary to open the refrigerator.

**Input：**

The input contains four lines. Each of the four lines contains four characters describing the initial state of appropriate handles. A symbol “+” means that the handle is in closed state, whereas the symbol “−” means “open”. At least one of the handles is initially closed.

**Output:**

The first line of the input contains N – the minimum number of switching. The rest N lines describe switching sequence. Each of the lines contains a row number and a column number of the matrix separated by one or more spaces. If there are several solutions, you may give any one of them.

**Sample input:**

-+--

----

----

-+--

**Sample output:**

6

1 1

1 3

1 4

4 1

4 3

4 4

**解题思路：**

先看一个简单的问题,如何把'+'变成'-'而不改变其他位置上的状态?答案是将该位置(i,j)及位置所在的行(i)和列(j)上所有的handle更新一次,结果该位置被更新了7次,相应行(i)和列(j)的handle被更新了4次,剩下的被更新了2次.被更新偶数次的handle不会造成最终状态的改变.因此得出高效解法,在每次输入碰到'+'的时候,自增该位置与相应的行和列,当输入结束后,遍历数组,所有为奇数的位置则是操作的位置,而奇数位置的个数之和则是最终的操作次数.

**Accepted Code：**

#include <iostream>

using namespace std;

int handles[4][4]={0};

int main ()

{

int step\_num=0;

for(int i=0;i<4;i++)

for(int j=0;j<4;j++)

{

char ch;

cin>>ch;

if(ch=='+') //输入+时，自增该位置与相应的行和列

{

handles[i][j]++;

for(int k=0;k<4;k++)

{

handles[i][k]++;

handles[k][j]++;

}

}

}

for(int i=0;i<4;i++)

for(int j=0;j<4;j++) //遍历数组

{

if(handles[i][j]%2) //如果为基数，step\_num加一

step\_num++;

}

cout<<step\_num<<endl; //输出步骤数

for(int i=0;i<4;i++)

for(int j=0;j<4;j++)

{

if(handles[i][j]%2) //输出操作的的位置

cout<<i+1<<' '<<j+1<<endl;

}

}