1213：八皇后问题

题目描述

在国际象棋棋盘上放置八个皇后，要求每两个皇后之间不能直接吃掉对方。

输入

无

输出

按给定顺序和格式输出所有八皇后问题的解（见样例）。

输入样例

无

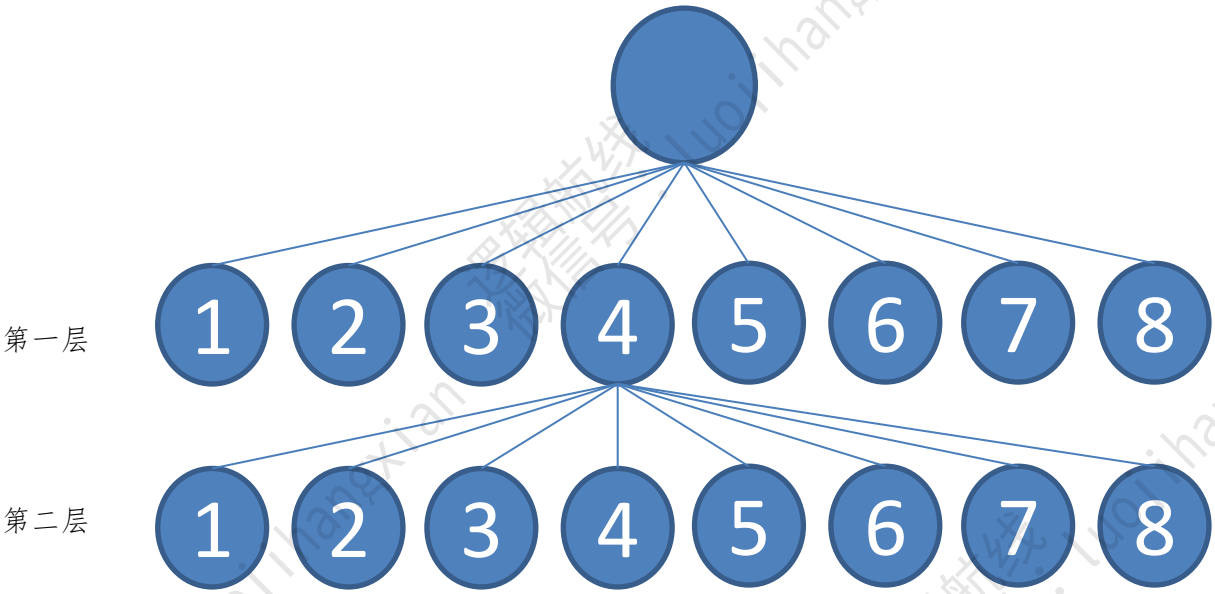
输出样例

```
No. 1
1 0 0 0 0 0 0 0
0 0 0 0 0 0 1 0
0 0 0 0 1 0 0 0
0 0 0 0 0 0 0 1
0 1 0 0 0 0 0 0
0 0 0 1 0 0 0 0
0 0 0 0 0 1 0 0
0 0 1 0 0 0 0 0
No. 2
1 0 0 0 0 0 0 0
0 0 0 0 0 0 1 0
0 0 0 1 0 0 0 0
0 0 0 0 0 1 0 0
0 0 0 0 0 0 0 1
0 1 0 0 0 0 0 0
0 0 0 0 1 0 0 0
0 0 1 0 0 0 0 0
...以下省略
```

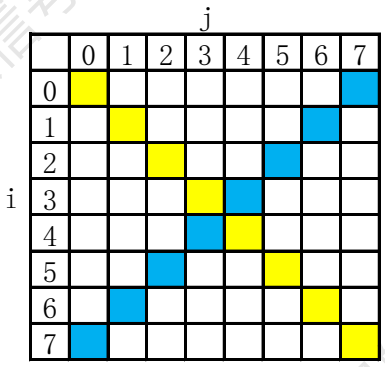
解析

本体的方案数非常容易想到，那就是每一层都从第一列开始尝试摆放，然后分别比较同行、同列、同对角线是否能够放置即可。

所以，我们可以构建如下决策树：



在这里有个小知识，那就是如何判断两个点在相同的斜线上？我们观察如下图形：



其中黄色的我们称为主对角线，通过观察我们发现，在主对角线上的点的特征是：横纵坐标的差相等，例如2-2 == 3-3。

对于蓝色的副对角线，它的特征是横纵坐标和相等，例如：7+0 == 1+6。

这两条规范可以推广到任意的斜线上。

编码

```
#include <bits/stdc++.h>

using namespace std;

int box[10];  //索引：当前的行号，值：当前的列号
int col = 8;  //一共的行列数
int total; //总共的方案数

void print() {
    printf("No. %d\n", total);
    //遍历每一列
    for (int i = 0; i < col; i++) {
        //遍历每一行
        for (int j = 0; j < col; j++) {
            //如果记录的列数和当前的列相同的时候，说明需要输出1
            if (box[j] == i) {
                printf("1 ");
            } else {
                printf("0 ");
            }
        }
        printf("\n");
    }
}

//检测当前行是否可以放置
bool Check(int row) {
    //遍历先前摆放过的行
    for (int j = 0; j < row; j++) {
        //判断是否在相同的列上，有则无法摆放
        bool colRes = (box[row] == box[j]);
        //判断主对角线是否存在皇后
        bool line1Res = (row - box[row] == j - box[j]);
        //判断副对角线是否存在皇后
        bool line2Res = (row + box[row] == j + box[j]);
        //不满足条件
        if (colRes || line1Res || line2Res) {
            return false;
        }
    }
    return true;
}

void dfs(int row) {
    //已经全部找到，可以进行打印了，一共有step列
```

```
    if (row == col) {
        //记录找到了一种方案数
        total++;
        print();
        return;
    }

    //总计8列
    for (int i = 0; i < col; i++) {
        //将当前的位置进行摆放
        box[row] = i;
        //当前位置可以摆放
        if (Check(row)) {
            //向下一个盒子进行放置
            dfs(row + 1);
        }
    }
}


int main(int argc, char **argv) {
    dfs(0);
    return 0;

}
```