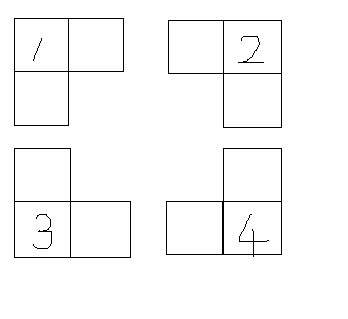
**残缺的棋盘**

有一正方形棋盘，其边长为2^k（1<k<10），如图a所示，其中有一格损坏。现在想用图b所示形状的硬纸板将没有坏的所有格子盖起来。而硬纸板不得放入坏格中和棋盘外。编程输出一种覆盖方案



输入: k  残缺格的坐标x,y（0<=x,y<2^k）

输出: 数字方阵,用7表示残缺的格

**代码：**

#define \_CRT\_SECURE\_NO\_WARNINGS 1

#include<stdio.h>

#include<stdlib.h>

int an[1100][1100];

void chess(int size, int x, int y, int posx, int posy);

int main()

{

int k, x, y;

scanf("%d%d%d", &k, &x, &y);

chess((1 << k), y, x, 0, 0);

an[y][x] = 7;

for (int i = 0; i<(1 << k); i++)

{

for (int j = 0; j<(1 << k); j++)

{

printf("%d ", an[i][j]);

}

if (1 != (1 << k) - 1)

putchar('\n');

}

system("pause");

return 0;

}

void chess(int size, int x, int y, int posx, int posy)

{

if (size == 1)

return;

int s = (size >> 1);

if (x<posx + s&&y<posy + s)

{

an[posx + s - 1][posy + s] = 4;

an[posx + s][posy + s - 1] = 4;

an[posx + s][posy + s] = 4;

chess(s, x, y, posx, posy);

chess(s, posx + s - 1, posy + s, posx, posy + s);

chess(s, posx + s, posy + s - 1, posx + s, posy);

chess(s, posx + s, posy + s, posx + s, posy + s);

}

else if (x<posx + s&&y >= posy + s)

{

an[posx + s - 1][posy + s - 1] = 3;

an[posx + s][posy + s - 1] = 3;

an[posx + s][posy + s] = 3;

chess(s, posx + s - 1, posy + s - 1, posx, posy);

chess(s, x, y, posx, posy + s);

chess(s, posx + s, posy + s - 1, posx + s, posy);

chess(s, posx + s, posy + s, posx + s, posy + s);

}

else if (x >= posx + s&&y<posy + s)

{

an[posx + s - 1][posy + s - 1] = 2;

an[posx + s - 1][posy + s] = 2;

an[posx + s][posy + s] = 2;

chess(s, posx + s - 1, posy + s - 1, posx, posy);

chess(s, posx + s - 1, posy + s, posx, posy + s);

chess(s, x, y, posx + s, posy);

chess(s, posx + s, posy + s, posx + s, posy + s);

}

else if (x >= posx + s&&y >= posy + s)

{

an[posx + s - 1][posy + s - 1] = 1;

an[posx + s - 1][posy + s] = 1;

an[posx + s][posy + s - 1] = 1;

chess(s, posx + s - 1, posy + s - 1, posx, posy);

chess(s, posx + s - 1, posy + s, posx, posy + s);

chess(s, posx + s, posy + s - 1, posx + s, posy);

chess(s, x, y, posx + s, posy + s);

}

}

结果示例：

