扫雷代码大概实现方法

首先用typedef弄个结构体变量Game

typedef struct gg

{

int a;//行

int b;//列

int c;//雷数

int \*\*seem\_1;//仅用于初始化，之后都不发生改变

int \*\*seem\_2;//用于用户操作之后的显示

}Game;

包括大概体函数void menu(Game \*tab)

{

printf("请按照提示进行游戏\n");

printf("请问你想玩多大规格的扫雷(n\*m)\n");

printf("请输入n和m\n");

scanf("%d%d", &tab->a, &tab->b);

tab->seem\_1 = space(tab->a, tab->b);--------这里用来申请空间

tab->seem\_2 = space(tab->a, tab->b);--------这里用来申请空间

tozero(tab->seem\_1, tab->a, tab->b);-------将申请的空间(结构体里的成员指向int类型指针的指针相当于一个数组)中的值全定义为0

tozero(tab->seem\_2, tab->a, tab->b); -------将申请的空间中的值全定义为0

printf("请输入雷的数量:\n");

scanf("%d", &tab->c);

nn(tab, tab->seem\_1, tab->a, tab->b, tab->c);----填入雷(设置两个随机变量充当填雷位置的坐标每次循环都生成不一样的随机数)

answer(tab);

printf("\n");

printf("\t\t\t\t\t\t\t游戏开始\n");

printf("\n");

eye(tab);

while (1)

{

yer(tab);

}

}

--------------------------------------------------------------------------------------------------------------------------------------------------------------

不知道怎么讲：讲一下流程吧！

提示用户输入扫雷规模(n\*m)和雷数(z)；然后创建两个(n\*m)大的空间（ int \*\*seem\_1;//仅用于初始化，int \*\*seem\_2;//用于用户操作之后的显示）并将其值全部赋值为0；循环z次创建随机坐标(a,b),0<a<n,0<b<m;把坐标(a,b)的位置赋值为1，即填入雷；完成基本准备。

可以打印出扫雷的答案了

提示用户输入坐标(x,y)判断坐标(x,y)的值（就是seem\_1）如果是1，游戏结束，为0，继续；

为0时判断周围8个格子的雷数，若其他格子上的值也为0继续显示周围的雷数，直到遇到格子的值为1时停止扩撒；程序通过标记p来确定雷当p的坐标和雷的坐标全部相等时，玩家胜利

@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@

扩散的思路：

用户输入坐标，判断坐标周围雷数，若雷数大于0不扩散，若雷数等于0，则依次判断周围8个格子的雷数，然后继续判断雷数，直到雷数大于零；就函数调用自身；