

## Lab1设计文档

主要算法思路：

```
int row=0;//地雷图的宽度——二维数组的行
int col=0;//地雷图的长度——二维数组的列
int number=0;//地雷的数量
```

```
void initial(){//展示个人信息并初始化地雷图的长度、宽度，以及拥有地雷的数量
    cout<<"Designed by 沈远哲 21302010011"<<endl;
    while(row<1||col<1||row*col<=number){
        cout<<"使用命令行输入三个正整数,分别表示地雷图的长度、宽度，以及拥有地雷的数量"<<endl;
        cin>>row>>col>>number;
    }
}
```

展示个人信息并初始化地雷图的长度、宽度，以及拥有地雷的数量。同时判断输入数据的可执行性。

```
void generate(char **map){//随机生成雷的位置
    for(int i=0; i<row; i++){
        for(int j=0; j<col; j++){
            map[i][j]='0';
        }
    }
    srand ((unsigned) time (NULL));
    int inumber=0;//雷已生成数量计数器
    while(inumber<number){
        int rm, cm;
        rm=rand()%row;
        cm=rand()%col;
        if(map[rm][cm]=='*') continue;
        else{
            map[rm][cm]='*';
            inumber++;
        }
    }
}
```

随机生成雷的位置。首先将二维数组全部赋0，而后逐个随机生成雷的位置，如果此位置没有雷，则雷已生成数量计数器inumber加一；否则直接跳至循环开头再

生成随机位置，直至生成指定数量的地雷。

```
void print(char **map){//打印地雷图方阵
    for(int i=0; i<row; i++){
        for(int j=0; j<col; j++){
            cout<<map[i][j]<<" ";
        }
        cout<<endl;
    }
}
```

按行逐个打印二维数组中的元素。

```
int main() {
    initial();
    char *map[col]; //生成二维数组并分配空间
    for(int i=0; i<row; i++){
        map[i]=new char [col];
    }
    generate(map);
    print(map);
}
```

生成二维数组并调用各函数

运行结果展示：



不正确输入会要求再次输入。



