

# Lab2

## 1: 运行结果

由于 lab1 没有对输入进行判断，故在 lab2 中补充。其中 s 0 0 是临时加的打开地雷图的一个特定操作。

```
c:\Users\21714\Desktop\C++\lab1\MineSweeper.exe
欢迎来到侯斌洋制作的扫雷游戏！
现在请输入雷区的长（小于24）、宽（小于30）和地雷数量
输入格式类似于:9 9 10
啊 a a
输入数据不满足要求，请重新输入。
c a 9
输入数据不满足要求，请重新输入。
1 2 c
输入数据不满足要求，请重新输入。
0 0 0
输入数据不满足要求，请重新输入。
9 9 100
输入数据不满足要求，请重新输入。
8 8 10
# # # # # # # #
# # # # # # # #
# # # # # # # #
# # # # # # # #
# # # # # # # #
# # # # # # # #
# # # # # # # #
# # # # # # # #
# # # # # # # #
s 0 0
1 2 * 2 1 0 0 0
1 * 3 * 1 0 0 0
1 1 3 3 4 2 1 0
0 0 1 * * * 1 0
0 0 1 2 3 2 2 1
1 1 0 0 0 0 2 *
* 1 0 0 1 1 3 *
1 1 0 0 1 * 2 1
请按任意键继续. . .
```

## 2：代码原理

地雷图数字只要对非地雷的空格周围 8 个（边界的话另算）空格的地雷数量进行判断。故用双层 for 循环遍历每一个空格，跳过有地雷的格子，然后对空格判断周围地雷数量并将数字放在该格子上。判断条件可以直接写八个 if 语句，也可以用循环来完成，下面给出了两种代码。

其中对于边界的判定，只需要在计算地雷数量的时候略过超出了边界的判断条件即可，具体参照以下代码。

## 3：代码附录

以下代码均经过测试，输出正确。

(1) 直接用 8 个 if 判断

```
for (int i = 0; i < m_length; i++)
{
    int count;
    for (int j = 0; j < m_width; j++)
    {
        if (m_map[i][j] == -1)
        {
            continue;
        }
        else
        {
            count = 0;
            {
                if (i - 1 >= 0 && j - 1 >= 0 && m_map[i - 1][j - 1] == -1)
                    count++;
                if (i - 1 >= 0 && m_map[i - 1][j] == -1)
                    count++;
                if (i - 1 >= 0 && j + 1 < m_width && m_map[i - 1][j + 1] == -1)
                    count++;
                if (j - 1 >= 0 && m_map[i][j - 1] == -1)
                    count++;
                if (j + 1 < m_width && m_map[i][j + 1] == -1)
                    count++;
                if (i + 1 < m_length && j - 1 >= 0 && m_map[i + 1][j - 1] == -1)
                    count++;
                if (i + 1 < m_length && m_map[i + 1][j] == -1)
                    count++;
                if (i + 1 < m_length && j + 1 < m_width && m_map[i + 1][j + 1] == -1)
                    count++;
            }
            m_map[i][j] = count;
        }
    }
}
```

(2) 借助循环判断

```
int x[3] = {-1, 0, 1};
int y[3] = {-1, 0, 1};
for (int i = 0; i < m_length; i++)
{
    int count;
    for (int j = 0; j < m_width; j++)
    {
        if (m_map[i][j] == -1)
        {
            continue;
        }
        else
        {
            count = 0;
            {
                for (int m = 0; m < 3; m++)
                {
                    for (int n = 0; n < 3; n++)
                    {
                        if (i + x[m] >= 0 && i + x[m] < m_length && j + y[n] >= 0
                            && j + y[n] < m_width && m_map[i + x[m]][j + y[n]] == -1)
                        {
                            count++;
                        }
                    }
                }
            }
            m_map[i][j] = count;
        }
    }
}
```