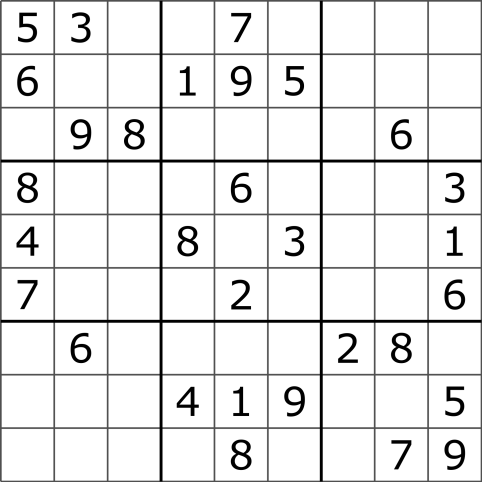
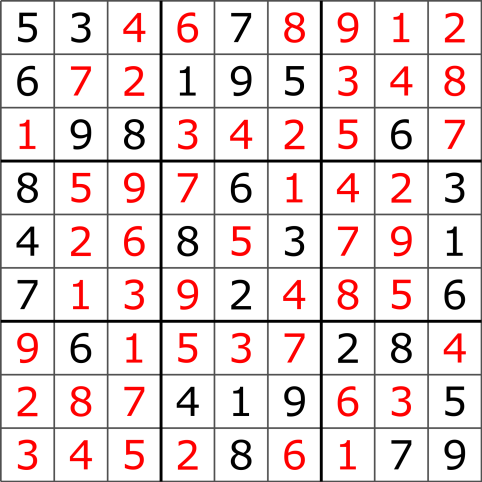
**C++基础与深度解析**

各位同学们，大家好，你们期盼已久的第二个项目终于来了，提前祝愿大家能顺利通过挑战~

**项目2：数独**

数独（sudoku）是是源自18世纪[瑞士](https://baike.baidu.com/item/%E7%91%9E%E5%A3%AB/131482" \t "https://baike.baidu.com/item/%E6%95%B0%E7%8B%AC/_blank)的一种数学游戏。玩家需要根据 9×9 盘面上的已知数字，推理出所有剩余空格的数字，并满足每一行、每一列、每一个粗线宫内的数字均含 1-9 ，且不能重复。

使用搜索求解数独问题，是算法问题经典中的经典，希望同学们可以编写C++程序来解决问题，巩固所学。一般来说，数独谜题答案是唯一的，但我们的程序完全可以处理多解的情况。

**具体要求：**

1.分别读入测试数据easy.txt和medium.txt，每组数据共9行9列，若数字为0则代表空格；

2.输出搜索的结果

2.1.对easy数据，输出数独的解，即一幅填好的数独；

2.2.对medium数据，输出解数；

3.输出运行时间，可以参考附件中的代码框架；

**4.在说明文档中展示搜索的结果、运行时间和剪枝的思路。**

**扩展：**

尝试求解hard，hard+，hard++这几份数据的解数。由于不同环境下程序运行时间会有差异，**我们作业的评判将重点关注算法实现过程（比如剪枝的思路）**，当然，**最终目标是希望hard++能够在不开启编译优化时还能秒出结果**，希望大家朝这个方向努努力。下面这份表格供大家参考。

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **难度** | **easy** | **medium** | **hard** | **hard+** | **hard++** |
| **解数** | **1** | **225** | **6592** | **15540** | **31322** |

**提示：**

1.数独的状态不方便存储，因此建议使用深度优先搜索（DFS）；

2.剪枝的小技巧有很多。比如，使用位运算来检查行/列/宫格的状态；也可以结合启发式搜索的思想，看看怎样才能够让后续搜索的状态减少；(8.4:目前我的算法还是属于DFS的穷举法)

3.舞蹈链算法（Dancing Links）对数独问题的求解有奇效。

优化思路：

1、无需记录解的形式，只需记录解的数量，即可以提高程序的运行速度。