

【注意:】 1、本次作业不允许使用后续课程中的指针等相关概念
2、不允许使用 C++ 的 string 变量!!!

补充:

7、题目同 5-b6, 要求数据从文件 score.dat 中读入, 且数据个数不限, 以-1 做为输入的结束

- 【要求:】** 1、假设数据总数不超过 1000 个, 且保证是[0-100]间的整数
2、输入结束标记-1 自身不计入有效数据内
3、给出示例的 5-b6-makedat.exe, 运行一次, 可生成一个 score.dat 文件

8、给定一个 9*9 的矩阵, 判断是否满足数独的解

- 【要求:】** 1、假设矩阵中填充的数据全部都是 1-9, 不必考虑非正常数据
2、数据从文件 shudu.txt 中读取
3、给出示例的 5-b7-maketxt.exe, 运行一次, 可生成一个 shudu.txt 文件

9、用函数+数组方式重新完成 3-b8(数字转人民币大写), 要求如下:

- 【要求:】** 1、所有的大写数字均放在全局一维字符数组 chistr 中, 具体形式为
char chistr[]="零\0 壹\0 贰\0 叁\0 肆\0 伍\0 陆\0 柒\0 捌\0 玖\0";
凡需输出“零-玖”的地方, **只允许**从此数组中取值
2、其它内容(拾佰仟万亿圆角分整)可自行组织输出
3、转换后的内容**不允许逐次输出**, 必须将所有内容依次放入一个全局字符数组 char result[256] 中, 最后的输出**只允许**用一句 cout << result<< endl 来完成
(除了输入数字时的提示信息及此句 cout 外, 程序中不允许出现其它 cout 语句)
4、保证 3-b8 中的所有测试数据均通过
【提示:】 1、根据分解的各位数字从 chistr 中取部分内容
2、各位数字要输出的内容依次放入 result 中, 最后输出这个字符串即可

10、生成并打印 Windows 扫雷游戏的内部数组结构

- 【Windows 扫雷游戏的玩法:】** 1、开始游戏, 以高级难度 16*30 的位置中 99 颗雷为例, 此时虽然屏幕无显示, 但 99 颗雷在什么位置内部已知
2、按下鼠标左键, 表示玩家确认该位置不是雷, 此时若其周围 8 个位置均无雷(四角位置: 1-3, 四边位置: 1-5, 下同), 则屏幕显示空白(会将所有相连的空白位置全部显示), 否则会按周围 8 个位置有几颗雷来显示数字 1-8; 如果该位置是雷, 则给出提示, 游戏结束
3、按下鼠标右键, 表示玩家确认该位置是雷, 此时屏幕会显示小红旗(如果玩家判断错误, 此处不应是雷, 会导致后续判断错误)

- 【要求:】** 1、在 26x10 的范围内随机产生 50 颗雷(若生成的位置已有雷, 则需要再次生成新位置)
2、其它非雷位置分别给出 0-8, 表示其周围 8 个位置的雷数
3、输出形式如下(5-b10-demo.exe)

```
0 0 0 1 * * 2 1 1 1 0 0 2 * 2 0 0 1 * 2 1 0 0 0 0 0
1 1 0 1 3 * 2 1 * 1 0 1 3 * 2 1 1 2 2 * 1 0 0 0 0 0
* 1 0 0 2 2 2 2 2 1 2 * 2 2 3 * 3 2 2 1 0 0 0 0 0
1 2 1 1 1 * 2 2 * 1 1 * 2 1 1 * * 3 * 1 0 0 1 1 1 0
1 2 * 3 3 2 2 * 3 2 2 2 3 2 2 3 4 4 2 1 0 0 1 * 1 0
* 2 2 * * 2 1 2 4 * 3 3 * * 1 1 * * 1 0 0 0 1 1 1 0
1 1 1 4 * 3 0 2 * * * 3 * 4 2 2 2 2 1 0 1 1 2 1 2 1
1 1 0 2 * 3 1 2 * 4 2 2 2 4 * 2 0 0 0 0 1 * 3 * 3 *
* 1 0 1 3 * 2 2 3 3 1 0 1 * * 3 1 1 1 1 2 2 4 * 3 1
1 1 0 0 2 * 2 1 * * 1 0 1 2 3 * 1 1 * 1 1 * 2 1 1 0
```

【提示:】1、生成指定范围内随机数的方法参考 4-b9

2、可用 int 型数组，0-8 代表非雷的数字，另用某个数字代表雷
也可用 char 型数组，'0'-'8' 表示雷，'*' 表示雷

【本次作业占平时成绩分数: 8】

【作业要求:】

- 1、5 月 5 日前网上提交本次作业，网址: <http://210.75.207.54:7200>
- 2、每题所占平时成绩的具体分值见网页
- 3、超过截止时间提交作业会自动扣除相应的分数，具体见网页上的说明