

软件工程设计用户手册

结对编程



用户手册：数独生成与求解控制台程序

成员：黄韵竹 2011064、邹胡诚钰 2013958

专业：信息安全与法学

指导教师：徐思涵

目录

1 程序概述:	2
1.1 任务总体目标:	2
1.2 任务要求:	2
2 程序演示:	3
2.1 命令格式:	3
2.2 运行示例:	4
2.3 程序设计思路:	5
3 代码质量分析:	6
4 单元测试与覆盖率统计:	7
4.1 单元测试:	7
4.2 设计用例统计覆盖率:	7

1 程序概述:

1.1 任务总体目标:

实现一个能够生成数独终局并能求解数独问题的控制台程序,该命令行程序能实现如下功能:

- 1、生成不重复的数独终局至文件
- 2、读取文件内的数独问题,求解并将结果输出到文件

1.2 任务要求:

- 1、采用 C++语言实现,运行环境为 64-bit Windows 10;
- 2、对输入的命令进行判断,实现不同的功能,对非法输入进行报错;
- 3、生成不重复的数独终局至文件;
- 4、读取文件内的数独问题,求解并将结果输出到文件;
- 5、代码经过质量分析并消除警告;
- 6、设计测试用例来覆盖主要功能,统计覆盖率;

7、生成数独：shudu.exe -c 100（具体参数见命令格式表）

8、求解数独：shudu.exe -s path_of_file

2 程序演示：

2.1 命令格式：

参数名字	参数含义	范围限制	用法示例
-c	需要的数独终局数量	1-1000000	shudu.exe -c 100 表示生成 100 个数独终局保存到 sudoku_finality.txt 文件中
-s	需要求解的数独棋盘的文件路径	绝对或相对路径	shudu.exe -s solve.txt 表示求解当前目录下的 solve.txt 文件中的数独，并将结果保存到 sudoku.txt
-n	需要的游戏数量	1-10000	shudu.exe -n 1000 表示生成 1000 个数独游戏到同目录 sudoku_finality.txt 文件中
-m	生成游戏的难度	1-3 1：简单 2：中等 3：困难	shudu.exe -n 1000 -m 1 表示生成 1000 个简单数独游戏到同目录 sudoku_finality.txt 文件中，只有 m 和 n 一起使用才认为参数无误，否则报错
-r	生成游戏中挖空得到数量范围	20-55	shudu.exe -n 20 -r 20~55 表示生成 20 个挖空数在 20-55 之间的数独游戏到同目录 sudoku_finality.txt 文件中，只有 r 和 n 一起使用才认为参数无误，否则报错
-u	生成游戏的解唯一		shudu.exe -n 20 -u 表示生成 20 个解唯一的数独游戏到同目录 sudoku_finality.txt 文件中，只有 u 和 n 一起使用

			才认为参数无误，否则报错
--	--	--	--------------

2.2 运行示例:

- 生成数独:

cmd 打开命令行窗口，输入命令: shudu.exe -c 100

```
C:\Windows\System32\cmd.exe
Microsoft Windows [版本 10.0.19045.3086]
(c) Microsoft Corporation。保留所有权利。

C:\Users\Lenovo\source\repos\shudu\x64\Debug>shudu.exe -c 100
over create finality

C:\Users\Lenovo\source\repos\shudu\x64\Debug>a_
```

生成数独终局文件在同目录下的 sudoku_finality.txt 文件:

ques.txt	2021/1/6 3:54	文本文档	169 KB
shudu.exe	2023/6/28 21:59	应用程序	185 KB
shudu.exe.lastcodeanalysisucceeded	2023/6/28 22:00	LASTCODEANA...	0 KB
shudu.pdb	2023/6/28 21:59	Program Debug ...	1,588 KB
solve.txt	2023/6/28 14:54	文本文档	169 KB
sudoku.txt	2023/6/28 20:31	文本文档	338 KB
sudoku_finality.txt	2023/6/28 22:39	文本文档	17 KB
UnitTest2.dll	2023/6/28 20:31	应用程序扩展	241 KB
UnitTest2.exp	2023/6/28 20:06	Exports Library ...	19 KB
UnitTest2.lib	2023/6/28 20:06	Object File Library	24 KB
UnitTest2.pdb	2023/6/28 20:31	Program Debug ...	2,700 KB

sudoku.txt - 记事本

文件(E) 编辑(E) 格式(O)

```
9 1 2 3 4 5 6 7 8|
6 7 8 9 1 2 3 4 5
3 4 5 6 7 8 9 1 2
1 2 3 4 5 6 7 8 9
7 8 9 1 2 3 4 5 6
4 5 6 7 8 9 1 2 3
8 9 1 2 3 4 5 6 7
5 6 7 8 9 1 2 3 4
2 3 4 5 6 7 8 9 1

9 1 2 3 4 5 6 8 7
6 8 7 9 1 2 3 4 5
3 4 5 6 8 7 9 1 2
1 2 3 4 5 6 8 7 9
8 7 9 1 2 3 4 5 6
4 5 6 8 7 9 1 2 3
7 9 1 2 3 4 5 6 8
5 6 8 7 9 1 2 3 4
2 3 4 5 6 8 7 9 1
```

其余生成数独命令类似，生成文件保存到同目录 `sudoku_finality.txt` 文件中，因此不再过多阐述。

● 求解数独：

cmd 打开命令行窗口，输入命令：`shudu.exe -s solve.txt`

```
C:\Users\Lenovo\source\repos\shudu\x64\Debug>shudu.exe -s solve.txt
over deal question
```

求解得到结果保存到同目录 `sudoku.txt` 文件：

文件(F) 编辑(E) 格式(O)	文件(E) 编辑(E) 格式(O) 查看(V)
4 1 2 3 0 0 0 0 8	4 1 2 3 9 6 7 5 8
0 5 0 0 0 0 4 0 2	3 5 6 7 1 8 4 9 2
0 0 0 0 0 0 0 0 0	7 9 8 4 5 2 3 1 6
0 2 0 0 6 0 0 8 0	1 2 4 5 6 3 9 8 7
0 6 7 0 8 4 1 2 3	5 6 7 9 8 4 1 2 3
9 8 0 1 0 0 5 6 0	9 8 3 1 2 7 5 6 4
2 0 5 6 7 9 8 0 1	2 3 5 6 7 9 8 4 1
6 0 0 8 0 0 2 0 5	6 4 9 8 3 1 2 7 5
0 0 1 0 0 5 0 0 0	8 7 1 2 4 5 6 3 9
4 1 0 0 5 0 7 0 8	4 1 2 3 5 6 7 9 8
3 5 0 0 9 0 0 0 0	3 5 6 7 9 8 4 1 2
7 9 8 0 0 2 3 0 6	7 9 8 4 1 2 3 5 6
0 0 3 0 6 7 0 8 4	1 2 3 5 6 7 9 8 4
5 0 0 0 0 0 1 2 0	5 6 7 9 8 4 1 2 3
0 8 4 0 2 3 5 6 7	9 8 4 1 2 3 5 6 7
0 3 5 0 7 9 8 4 1	2 3 5 6 7 9 8 4 1
8 0 1 0 3 0 0 7 0	8 4 1 2 3 5 6 7 9
0 0 9 8 4 1 0 3 0	6 7 9 8 4 1 2 3 5

2.3 程序设计思路：

● 求解思路：

求解思路为深度优先搜索。主要优化方式是调整搜索顺序，首先通过遍历数独得到各行各列各 9 宫格的空格数，同时统计空格数可以填的数的数量（因为有行列和九宫格的限制），然后进行排序，从数量较小的空格开始遍历。可以减小搜索的时间。

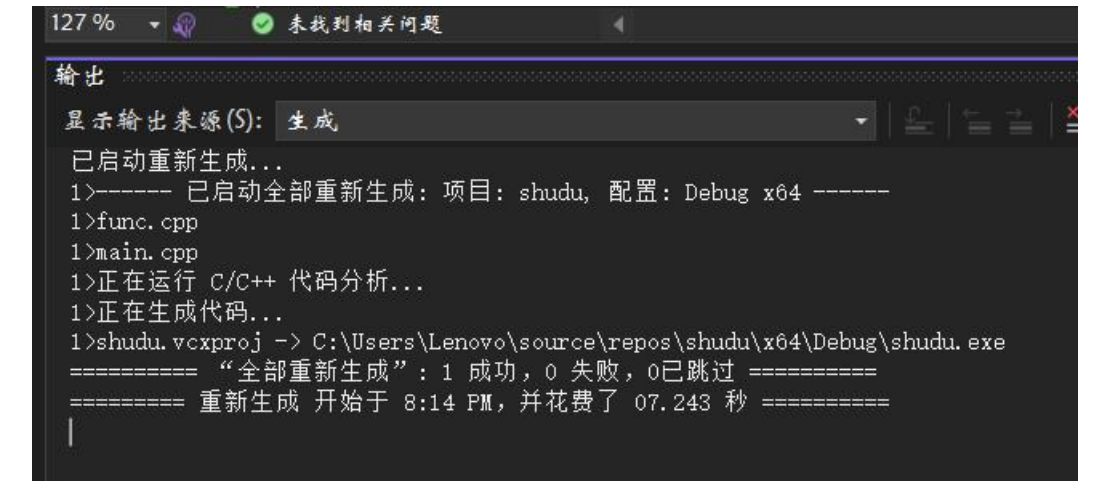
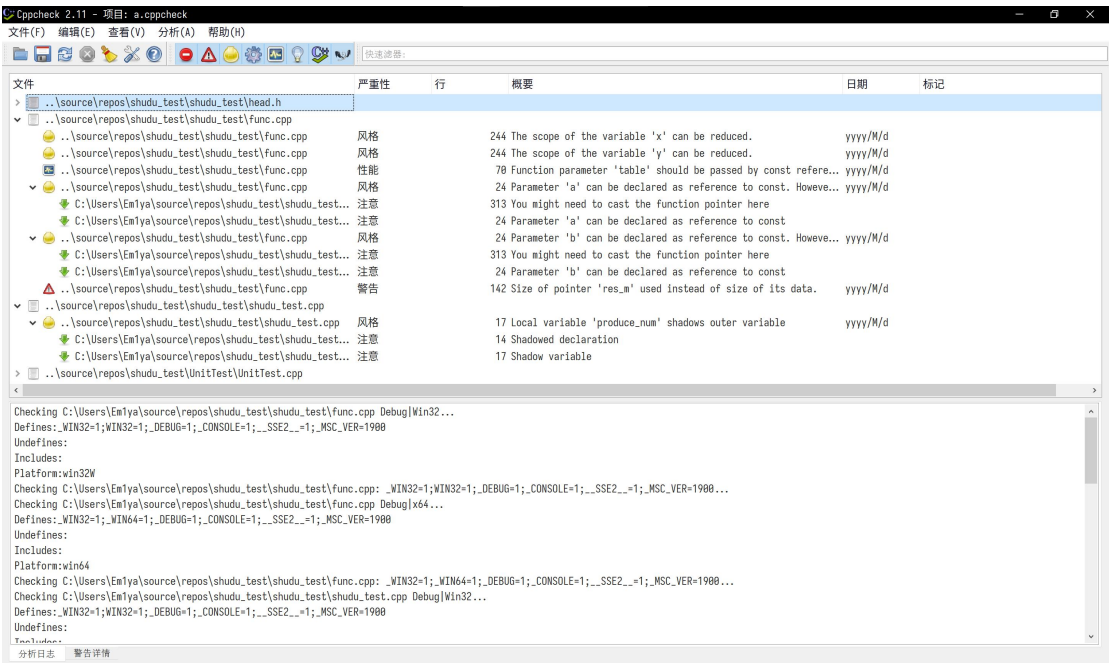
● 生成终局思路：

由数独规则可知，数独终局在成型之后数字的交换不会破坏数独完整性（比

如把 8 和 9 全部交换），因此可以通过数字交换获得数独终局。共 8！种方法。其次，数独 1-3，4-6，7-9 行的交换也不会造成数独终局被破坏。因此使用行之间的交换也是一种思路。共有 3！*3！*2！种交换。¹

3 代码质量分析：

利用 cppcheck 进行代码质量分析，并消除所有警告：

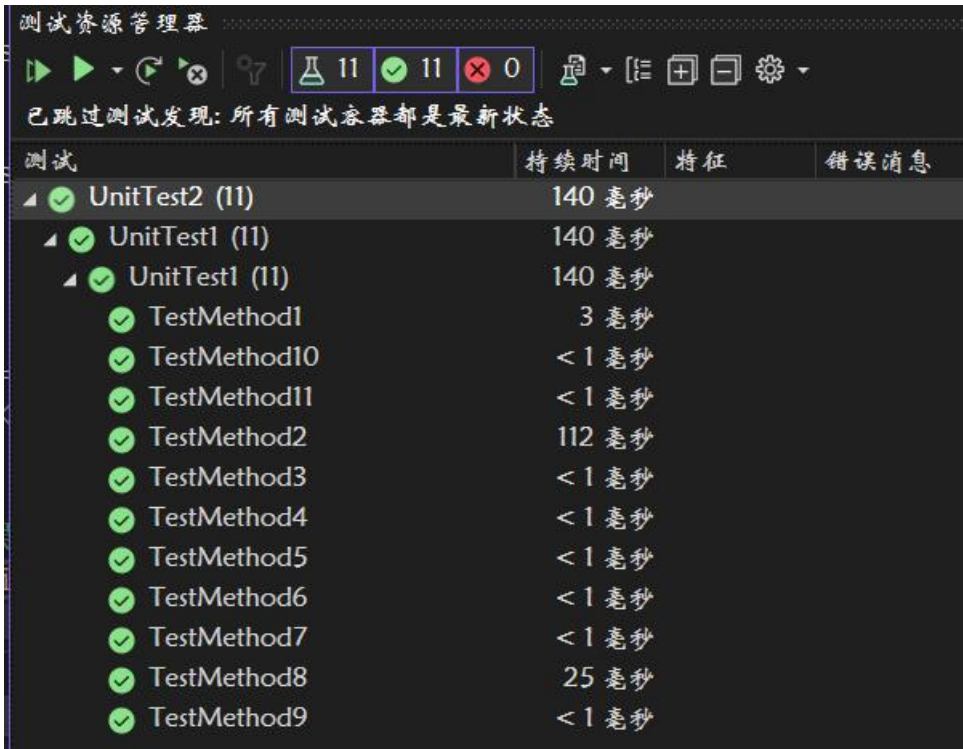


¹ 参考 csdn 博主雨暮《数独终局生成和求解控制台程序（c++）》

4 单元测试与覆盖率统计:

4.1 单元测试:

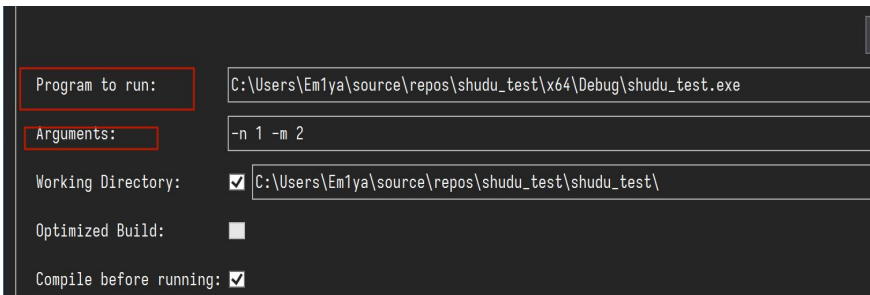
单元测试代码详见代码文件，主要对越界值、正常值、九宫格位置、路径、文件格式、dfs 返回等进行测试，单元测试结果如下：



测试	持续时间	特征	错误消息
✔ UnitTest2 (11)	140 毫秒		
✔ UnitTest1 (11)	140 毫秒		
✔ UnitTest1 (11)	140 毫秒		
✔ TestMethod1	3 毫秒		
✔ TestMethod10	< 1 毫秒		
✔ TestMethod11	< 1 毫秒		
✔ TestMethod2	112 毫秒		
✔ TestMethod3	< 1 毫秒		
✔ TestMethod4	< 1 毫秒		
✔ TestMethod5	< 1 毫秒		
✔ TestMethod6	< 1 毫秒		
✔ TestMethod7	< 1 毫秒		
✔ TestMethod8	25 毫秒		
✔ TestMethod9	< 1 毫秒		

4.2 设计用例统计覆盖率:

本程序使用的是 OpenCppCoverage 进行代码覆盖率统计，在 VS2019 中安装 OpenCppCoverage 插件，在工具中打开设置并在窗口设置命令行参数：



具体测试用例与覆盖率如下：

- 用例 1 : -c 123

Coverage

Filter: ☒ Display coverage

	Coverage	Covered line	Uncovered line	Total line
shudu_test.exe	15%	46	265	311
C:\Users\Em1ya\source\repos\shudu_test\x64\Debug\shudu_test.exe	15%	46	265	311
C:\Users\Em1ya\source\repos\shudu_test\shudu_test\func.cpp	13%	36	237	273
C:\Users\Em1ya\source\repos\shudu_test\shudu_test\shudu_test.cpp	26%	18	28	38

● 用例 2: -c 1000001

Coverage

Warning: Your program has exited with error code: -1

Filter: ☒ Display coverage

	Coverage	Covered line	Uncovered line	Total line
shudu_test.exe	8%	26	285	311
C:\Users\Em1ya\source\repos\shudu_test\x64\Debug\shudu_test.exe	8%	26	285	311
C:\Users\Em1ya\source\repos\shudu_test\shudu_test\func.cpp	6%	17	256	273
C:\Users\Em1ya\source\repos\shudu_test\shudu_test\shudu_test.cpp	24%	9	29	38

● 用例 3: -s C:\Users\Em1ya\source\repos\shudu_test\x64\Debug\ques.txt

Coverage

Filter: ☒ Display coverage

	Coverage	Covered line	Uncovered line	Total line
shudu_test.exe	31%	96	215	311
C:\Users\Em1ya\source\repos\shudu_test\x64\Debug\shudu_test.exe	31%	96	215	311
C:\Users\Em1ya\source\repos\shudu_test\shudu_test\func.cpp	18%	7	31	38
C:\Users\Em1ya\source\repos\shudu_test\shudu_test\shudu_test.cpp	33%	89	184	273

● 用例 4: -n 10

Coverage

Filter: ☒ Display coverage

	Coverage	Covered line	Uncovered line	Total line
shudu_test.exe	39%	121	198	311
C:\Users\Em1ya\source\repos\shudu_test\x64\Debug\shudu_test.exe	39%	121	198	311
C:\Users\Em1ya\source\repos\shudu_test\shudu_test\func.cpp	32%	12	26	38
C:\Users\Em1ya\source\repos\shudu_test\shudu_test\shudu_test.cpp	48%	189	164	273

● 用例 5: -n 1 -r 25

Coverage

Filter: ☒ Display coverage

	Coverage	Covered line	Uncovered line	Total line
shudu_test.exe	48%	123	188	311
C:\Users\Em1ya\source\repos\shudu_test\x64\Debug\shudu_test.exe	48%	123	188	311
C:\Users\Em1ya\source\repos\shudu_test\shudu_test\func.cpp	39%	187	166	273
C:\Users\Em1ya\source\repos\shudu_test\shudu_test\shudu_test.cpp	42%	16	22	38

● 用例 6: -n 10 -m 1

Coverage

Filter: ☒ Display coverage

	Coverage	Covered line	Uncovered line	Total line
shudu_test.exe	48%	123	188	311
C:\Users\Em1ya\source\repos\shudu_test\x64\Debug\shudu_test.exe	48%	123	188	311
C:\Users\Em1ya\source\repos\shudu_test\shudu_test\func.cpp	39%	187	166	273
C:\Users\Em1ya\source\repos\shudu_test\shudu_test\shudu_test.cpp	42%	16	22	38

● 用例 7: -n 10 -m 2

Coverage

Filter: ☒ Display coverage

	Coverage	Covered line	Uncovered line	Total line
shudu_test.exe	48%	125	186	311
C:\Users\Em1ya\source\repos\shudu_test\x64\Debug\shudu_test.exe	48%	125	186	311
C:\Users\Em1ya\source\repos\shudu_test\shudu_test\func.cpp	48%	189	164	273
C:\Users\Em1ya\source\repos\shudu_test\shudu_test\shudu_test.cpp	42%	16	22	38

● 用例 8: -n 1 -m 3

Coverage

Filter: ☒ Display coverage

	Coverage	Covered line	Uncovered line	Total line
shudu_test.exe	<div><div></div></div> 40%	124	187	311
C:\Users\Emiya\source\repos\shudu_test\x64\Debug\shudu_test.exe	<div><div></div></div> 40%	124	187	311
C:\Users\Emiya\source\repos\shudu_test\shudu_test\func.cpp	<div><div></div></div> 40%	108	165	273
C:\Users\Emiya\source\repos\shudu_test\shudu_test\shudu_test.cpp	<div><div></div></div> 42%	16	22	38

● 用例 9: -n 20 -u

Coverage

Filter: ☒ Display coverage

	Coverage	Covered line	Uncovered line	Total line
shudu_test.exe	<div><div></div></div> 40%	124	187	311
C:\Users\Emiya\source\repos\shudu_test\x64\Debug\shudu_test.exe	<div><div></div></div> 40%	124	187	311
C:\Users\Emiya\source\repos\shudu_test\shudu_test\func.cpp	<div><div></div></div> 40%	108	165	273
C:\Users\Emiya\source\repos\shudu_test\shudu_test\shudu_test.cpp	<div><div></div></div> 42%	16	22	38

● 用例 10: -n 20 -u 50

Coverage

Filter: ☒ Display coverage

	Coverage	Covered line	Uncovered line	Total line
shudu_test.exe	<div><div></div></div> 40%	125	186	311
C:\Users\Emiya\source\repos\shudu_test\x64\Debug\shudu_test.exe	<div><div></div></div> 40%	125	186	311
C:\Users\Emiya\source\repos\shudu_test\shudu_test\func.cpp	<div><div></div></div> 40%	109	164	273
C:\Users\Emiya\source\repos\shudu_test\shudu_test\shudu_test.cpp	<div><div></div></div> 42%	16	22	38