结对编程实验报告

简介

本文是南开大学《软件工程》课程中"结对编程"作业的实验报告，小组成员包括：

- 郤铭恺，2012411，计算机学院，计算机科学与技术专业，徐思涵老师班
- 刘宇航，2012232，计算机学院，计算机科学与技术专业，李起成老师班

报告内容包括简单的代码介绍，单元测试部分，简单用户手册，质量分析截图和测试覆盖率截图。下面将开始报告的主体部分。

简易代码介绍

我们代码的目录结构如下：

```
- build.sh
- case1.txt
- case2.txt
- CMakeLists.txt
+ include (dir)
  - api.hpp      （前后端耦合API）
  - clipp.h      （参数转换工具）
  - common.hpp   （必备的全局变量声明、宏定义和头文件）
  - gen.hpp      （生成数独相关函数）
  - parse.hpp    （参数处理相关函数）
  - sol.hpp      （数独求解相关函数）
  - sudoku.hpp   （数独类及方法声明）
- main.cc        （程序入口点）
- Makefile
- README.md
+ src (dir)
  - api.cc
  - common.cc
  - gen.cc
  - parse.cc
  - sol.cc
  - sudoku.cc
- test
- test.txt
```

其中，main.cc函数为程序的入口点，在include和src目录下定义着我们的代码主体。每个与hpp文件对应的cc文件装载着hpp中文件声明的定义。

我们代码以命令行作为前端，纯C++语言作为后端，前后端交互的方式通过主函数中对参数的处理代码调用api文件中的参数处理函数来进行。

单元测试

TODO

用户手册

使用说明

我们的程序使用CMake进行编译，具有**跨平台**的优良特性，运行方式为使用命令行后跟参数的方式运行，支持的参数列表和功能如下：

参数名称	参数含义	范围限制	用法示例
-c	需要生成的数独终盘数量	1-1000000	示例:sudoku.exe -c 20 [表示生成20个数独终盘]
-s	需要解的数独棋盘文件路径	绝对或相对路径	示例:sudoku.exe -s game.txt [表示从game.txt读取若干个数独游戏,并给出其解答,生成到sudoku.txt中
-n	需要的游戏数量	1-10000	示例:sudoku.exe -n 1000 [表示生成1000个数独游戏]
-m	生成游戏的难度	1，2，3	示例:sudoku.exe -n 1000 -m 1 [表示生成1000个简单数独游戏,只有m和n一起使用才认为参数无误,否则请报错
-r	生成游戏中挖空的数量范围	20-55	示例:sudoku.exe -n 20 -r 20~55 [表示生成20个挖空数在20到55之间的数独游戏,只有r和n一起使用才认为参数无误,否则请报错
-u	生成游戏的解唯一	没有参数	示例:sudoku.exe -n 20 -u[表示生成20个解唯一的数独游戏,只有u和n一起使用才认为参数无误,否则请报错

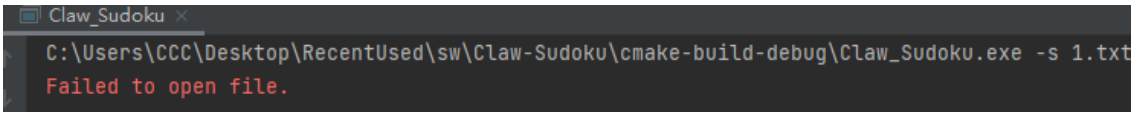
运行示例

TODO

注意事项

程序实现与测试中，我们为其约定了一些不可完成的任务。为了避开这些任务，并且避开一些意想不到的错误，我们提出一些输入约束与注意事项如下：

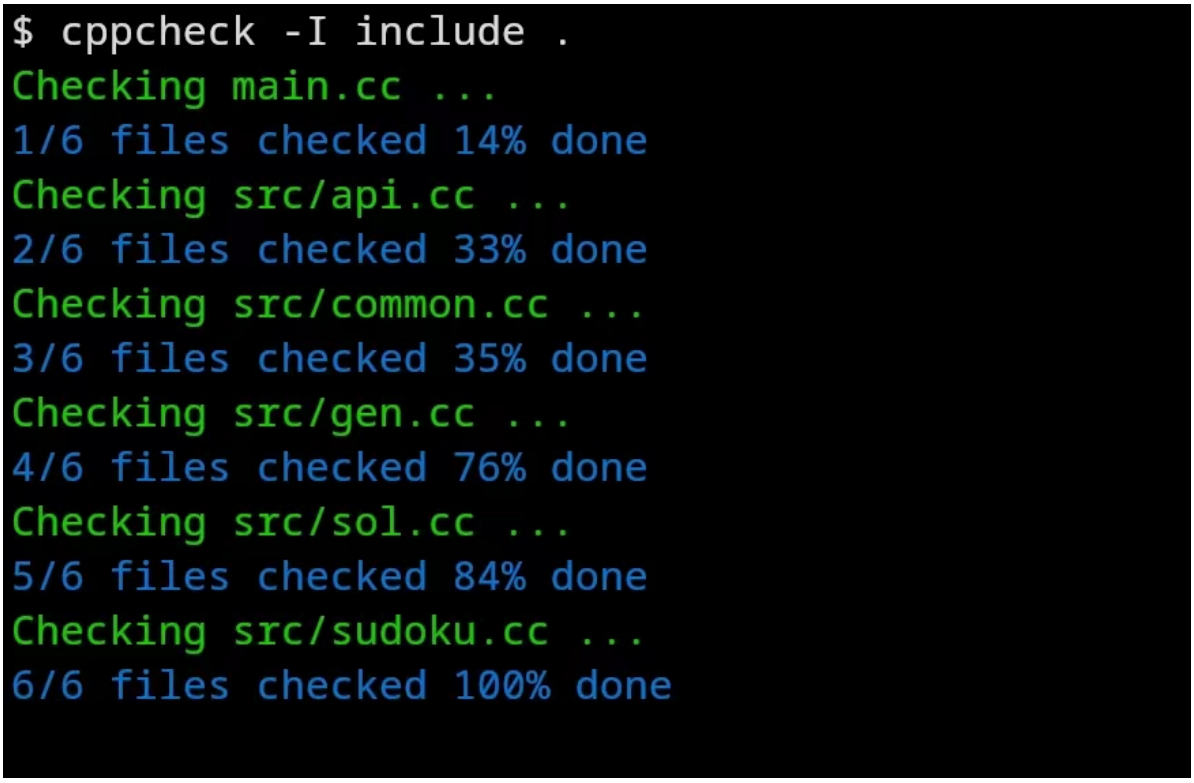
- 使用-c、-n、-m、-r输入参数时，务必确保输入在我们规定的范围内，并且请输入正确类型的参数（int或`<min>~<max>`）否则程序会提示assertion failed；
- 使用-r输入参数时，务必确保输入的格式正确（`<min>~<max>`），否则程序无法正确处理参数；
- 使用-s输入参数时，务必确保输入的文件路径正确且可以被访问，否则报错：



- 使用-m、-r、-u选项时，请务必确保配合-n选项使用；
- 在使用-n参数时，请选择合适的难度范围、挖空数量、生成数量进行输入，否则程序运行时间可能会过长，并且会耗费较多的CPU资源。

质量分析

我们使用cppcheck对代码进行静态分析，最终的分析结果如下：



质量分析全部通过，保证了我们代码编写的简洁易用性，并排除了潜在问题。

测试覆盖率

TODO