

南京大学
计算机 科学
与技术系
软件工程实验报告

实验名称：等价确认 GUI 工具

学 号： 201830204

姓 名： 顾秋涵

指导教师： 王豫

实验地点： 南京大学

实验时间： 2022.11

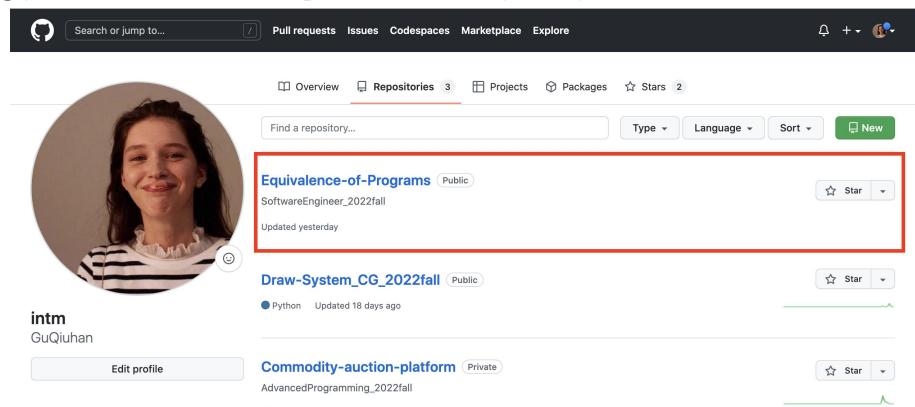
一、Git 操作

1. Github 仓库 (同实验四)

仓库地址:

HTTPS: <https://github.com/GuQiuhan/Equivalence-of-Programs.git>

SSH: git@github.com:GuQiuhan/Equivalence-of-Programs.git



The screenshot shows a GitHub user profile for 'intm GuQiuhan'. The profile picture is a photo of a young woman with dark hair. Below the profile picture, the GitHub handle 'intm' and name 'GuQiuhan' are displayed. There are three repositories listed:

- Equivalence-of-Programs** (Public)
SoftwareEngineer_2022fall
Updated yesterday
- Draw-System_CG_2022fall** (Public)
Python Updated 18 days ago
- Commodity-auction-platform** (Private)
AdvancedProgramming_2022fall

2. 添加、提交、远程提交文件

a. 提交前使用 git status 查看状态:

```
pro@prodeMacBook-Pro-2 SoftwareProj % git status
on branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    SoftwareProj.pro
    SoftwareProj.pro.user
    file.cpp
    file.h
    handle.cpp
    handle.h
    input/
    login.cpp
    login.h
    login.ui
    main.cpp
    mainwindow.cpp
    mainwindow.h
    mainwindow.ui
    user.cpp
    user.h
    users.txt

nothing added to commit but untracked files present (use "git add" to track)
```

b. 使用 git add 和 git commit 进行提交

```
[pro@prodeMacBook-Pro-2 SoftwareProj % git add .
[pro@prodeMacBook-Pro-2 SoftwareProj % git commit -m "control"
[master (root-commit) b428c53] control
 41 files changed, 1808 insertions(+)
 create mode 100644 SoftwareProj.pro
 create mode 100644 SoftwareProj.pro.user
 create mode 100644 file.cpp
 create mode 100644 file.h
 create mode 100644 handle.cpp
 create mode 100644 handle.h
 create mode 100644 input/4A/101036360.cpp
 create mode 100644 input/4A/117364748.cpp
 create mode 100644 input/4A/127473352.cpp
 create mode 100644 input/4A/134841308.cpp
 create mode 100644 input/4A/173077807.cpp
 create mode 100644 input/4A/48762087.cpp
 create mode 100644 input/4A/84822638.cpp
 create mode 100644 input/4A/84822639.cpp
 create mode 100644 input/4A/stdin_format.txt
 create mode 100644 input/50A/138805414.cpp
 create mode 100644 input/50A/142890373.cpp
 create mode 100644 input/50A/164831265.cpp
 create mode 100644 input/50A/21508887.cpp
 create mode 100644 input/50A/21508898.cpp
 create mode 100644 input/50A/21715601.cpp
 create mode 100644 input/50A/29019948.cpp
 create mode 100644 input/50A/30534178.cpp
 create mode 100644 input/50A/31034693.cpp
 create mode 100644 input/50A/33794240.cpp
 create mode 100644 input/50A/36641065.cpp
 create mode 100644 input/50A/45851050.cpp
 create mode 100644 input/50A/input.txt
 create mode 100644 input/50A/stdin_format.txt
 create mode 100644 input/equal.csv
 create mode 100644 input/unequal.csv
 create mode 100644 login.cpp
 create mode 100644 login.h
 create mode 100644 login.ui
 create mode 100644 main.cpp
 create mode 100644 mainwindow.cpp
 create mode 100644 mainwindow.h
 create mode 100644 mainwindow.ui
 create mode 100644 user.cpp
 create mode 100644 user.h
 create mode 100644 users.txt
```

- c. 提交后再次使用 git status 查看状态

```
[pro@prodeMacBook-Pro-2 SoftwareProj % git status
On branch master
nothing to commit. working tree clean
```

- d. 将本地仓库远程 push 到 github

```
[pro@prodeMacBook-Pro-2 SoftwareProj % git push -f origin master
Enumerating objects: 43, done.
Counting objects: 100% (43/43), done.
Delta compression using up to 8 threads
Compressing objects: 100% (36/36), done.
Writing objects: 100% (43/43), 13.17 KiB | 2.20 MiB/s, done.
Total 43 (delta 1), reused 0 (delta 0)
remote: Resolving deltas: 100% (1/1), done.
To github.com:GuQiuhan/Equivalence-of-Programs.git
 * [new branch]      master -> master
```

3. 在分支上进行开发

```
pro@prodeMacBook-Pro-2 SoftwareProj % git checkout control
M      SoftwareProj.pro.user
M      handle.cpp
D      input/equal.csv
D      input/unequal.csv
M      main.cpp
Switched to branch 'control'                                     切换分支开发

pro@prodeMacBook-Pro-2 SoftwareProj % git status
On branch control
Changes not staged for commit:                                查看状态
  (use "git add/rm <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
    modified:   SoftwareProj.pro.user
    modified:   handle.cpp
    deleted:    input/equal.csv
    deleted:    input/unequal.csv
    modified:   main.cpp

Untracked files:
  (use "git add <file>..." to include in what will be committed)
    exec_out/
    input/4A/input.txt
    output/

no changes added to commit (use "git add" and/or "git commit -m")
```

```
pro@prodeMacBook-Pro-2 SoftwareProj % git add .
[pro@prodeMacBook-Pro-2 SoftwareProj % git status
On branch control                                              将所有修改过的算
Changes to be committed:                                      法模块加入
  (use "git restore --staged <file>..." to unstage)       control分支中
    modified:   SoftwareProj.pro.user
    new file:  exec_out/4A/101036360
    new file:  exec_out/4A/101036360_output.txt
    new file:  exec_out/4A/117364748
    new file:  exec_out/4A/117364748_output.txt
    new file:  exec_out/4A/127473352
    new file:  exec_out/4A/127473352_output.txt
    new file:  exec_out/4A/134841308
    new file:  exec_out/4A/134841308_output.txt
    new file:  exec_out/4A/173077807
```

```
pro@prodeMacBook-Pro-2 SoftwareProj % git push -f origin control
Enumerating objects: 38, done.
Counting objects: 100% (38/38), done.                         commit后将分支push到远程仓库中
Delta compression using up to 8 threads
Compressing objects: 100% (25/25), done.
Writing objects: 100% (32/32), 12.67 KiB | 1.81 MiB/s, done.
Total 32 (delta 17), reused 0 (delta 0)
remote: Resolving deltas: 100% (17/17), completed with 5 local objects.
remote:
remote: Create a pull request for 'control' on GitHub by visiting:
remote:     https://github.com/GuQiuhan/Equivalence-of-Programs.git
remote:
To github.com:GuQiuhan/Equivalence-of-Programs.git
 * [new branch]      control -> control
pro@prodeMacBook-Pro-2 SoftwareProj %
```

4. 使用 git log 查看日志

```

pro@prodeMacBook-Pro-2 SoftwareProj % git log
commit 9489329e6673b22ee300b0369b041bf703ee5519 (HEAD -> master, origin/master)
Author: GuQiuhan <1650100563@qq.com>
Date: Tue Nov 22 20:47:38 2022 +0800

    addition

commit 503537adc99cc7f4fa99b578c434507e8997871e
Author: GuQiuhan <1650100563@qq.com>
Date: Tue Nov 22 19:29:48 2022 +0800

    finish

commit 4be4307548cca310b007fdb78ea1be3a98a7ad1b
Author: GuQiuhan <1650100563@qq.com>
Date: Tue Nov 22 11:53:43 2022 +0800

    11.22 v1

commit 98d723efabf5fdc10b62f812ee67542bee3eae8 (origin/control, control)
Author: GuQiuhan <1650100563@qq.com>
Date: Sun Nov 13 11:52:09 2022 +0800

    add command line & add output dir
    ....skipping...
commit 9489329e6673b22ee300b0369b041bf703ee5519 (HEAD -> master, origin/master)
Author: GuQiuhan <1650100563@qq.com>
Date: Tue Nov 22 20:47:38 2022 +0800

    addition

commit 503537adc99cc7f4fa99b578c434507e8997871e
Author: GuQiuhan <1650100563@qq.com>
Date: Tue Nov 22 19:29:48 2022 +0800

    finish

commit 4be4307548cca310b007fdb78ea1be3a98a7ad1b
Author: GuQiuhan <1650100563@qq.com>
Date: Tue Nov 22 11:53:43 2022 +0800

    11.22 v1

```

二、Git 其他命令操作 (扩展)

1. merge 和 rebase 的区别 (参考教程: <https://developer.aliyun.com/article/652579>)

merge 是合并的意思, rebase 是复位基底的意思。现在我们有这样的两个分支 test 和 master, 提交如下:



在 master 执行 git merge test 然后会得到如下结果:



在 master 执行 git rebase test,然后得到如下结果:



可以看到 merge 操作会生成一个新的节点,之前提交分开显示。而 rebase 操作不会生成新的

节点，是将两个分支融合成一个线性的操作。由此，想要更好的提交树，使用 rebase 操作会更好一点，这样可以线性的看到每一次提交，并且没有增加提交节点。此外，在操作中，merge 操作遇到冲突时候，当前 merge 不能继续下去。手动修改冲突内容后，add 修改，commit 就可以了。而 rebase 操作的话，会中断 rebase，同时会提示去解决冲突。解决冲突后，将修改 add 后执行 git rebase -continue 继续操作，或者 git rebase -skip 忽略冲突。

2. reset 和 revert 的区别 (参考教程: <https://www.51cto.com/article/678497.html>)

git revert 是用一次新的 commit 来回滚之前的 commit，git reset 是直接删除指定的 commit。

git reset 是把 HEAD 向后移动了一下，而 git revert 是 HEAD 继续前进，只是新的 commit 的内容和要 revert 的内容正好相反，能够抵消要被 revert 的内容。

在回滚这一操作上看，效果差不多。但是在日后的继续 merge 以前的老版本时有区别。git revert 是用一次逆向的 commit“中和”之前的提交，因此日后合并老的 branch 时，之前提交合并的代码仍然存在，导致不能够重新合并；但是 git reset 是之间把某些 commit 在某个 branch 上删除，因而和老的 branch 再次 merge 时，这些被回滚的 commit 应该还会被引入。

如果回退分支的代码以后还需要的情况则使用 git revert，如果分支是提错了没用的并且不想让别人发现这些错误代码，则使用 git reset。

```
pro@prodeMacBook-Pro-2 SoftwareProj % git log
[REDACTED] Commit 300723c7a0f37a10002f012cc07342bcc5ace0 (HEAD -> control, origin/master, origin/control, master)
Author: GuQiuhan <1650100563@qq.com>
Date:   Sun Nov 13 11:52:09 2022 +0800          git log 查看提交记录以及每次提交ID

    add command line & add output dir

commit 6428c53a4c60f6cd6dcd8d1fb4c3daa1f1efd21b (data, UI)
Author: GuQiuhan <1650100563@qq.com>
Date:   Sat Nov 12 17:19:19 2022 +0800

control
pro@prodeMacBook-Pro-2 SoftwareProj % git reset
pro@prodeMacBook-Pro-2 SoftwareProj % git log
[REDACTED] Commit 300723c7a0f37a10002f012cc07342bcc5ace0 (HEAD -> control, origin/master, origin/control, master)
Author: GuQiuhan <1650100563@qq.com>
Date:   Sun Nov 13 11:52:09 2022 +0800          使用git reset回退至上一次

    add command line & add output dir

commit 6428c53a4c60f6cd6dcd8d1fb4c3daa1f1efd21b (data, UI)
Author: GuQiuhan <1650100563@qq.com>
Date:   Sat Nov 12 17:19:19 2022 +0800          再次查看log

control
```

3. stash, cherry-pick 的使用等

● stash 储存操作

(参考教程:

<https://www.git-tower.com/learn/git/ebook/cn/command-line/branching-merging/stashing#start>) :

储藏 (Stash): 可以把储藏想象成一种剪贴板，它会获取你工作副本 (working copy) 中的所有改动，并且保存到一个新的剪贴板上。然后你就会得到一个“干净”的工作副本，也就是说一个不存在任何改动的工作目录。之后你随时都可以重新调回那些保存在剪贴板中的改动到你的工作副本中来，从而继续你之前没有完成的工作。你可以建立多个储藏单元，不仅仅局限于存储一组变化。同样，储藏也会不绑定在你所处的当前分支或是任何其它分支上，如果你想要调回任意一个储藏单元，它的改动将会被应用在你当前的 HEAD 分支上。

```
[REDACTED] pro@prodeMacBook-Pro-2 SoftwareProj % git stash
[REDACTED] No local changes to save
[REDACTED] pro@prodeMacBook-Pro-2 SoftwareProj % git stash list
```

- cherry-pick 有选择改动操作:

(参考教程: <https://www.ruanyifeng.com/blog/2020/04/git-cherry-pick.html>)

对于多分支的代码库, 将代码从一个分支转移到另一个分支是常见需求。这时分两种情况。一种情况是, 你需要另一个分支的所有代码变动, 那么就采用合并 (git merge)。另一种情况是, 你只需要部分代码变动 (某几个提交), 这时可以采用 Cherry pick。

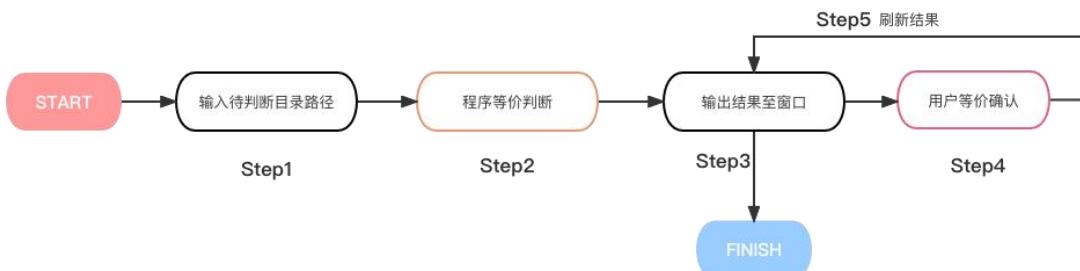
三、 代码设计

1. 运行流程

a. 程序等价判断流程:



b. 程序等价+程序确认判断流程:



2. 程序等价判断模块代码

a. 封装:

相比于实验四, 实验五加入了 GUI 模块, 所以将实验四中的 handle() 等处理程序等价的函数封装成一个 **函数对象类**, 通过调用 class Handle 中的 public 接口来处理输入的目录路径。

b. 具体实现:

a) 函数对象类:

```

13  class Handle
14  {
15  private:
16      QString dir;
17      vector<pair<QString,QString>> ansEqual;//存储结果,可直接访问
18      vector<pair<QString,QString>> ansUnequal;
19
20
21      vector<string> getFilesList(string dir); //读取目录列表
22      vector<string> getDirsList(string dirpath); //读取某一目录下的文件
23      string inputPara(string dir, string path); //构造随机输入
24      string exec(string dir, string input, string cpp);
25
26      void Classify(string Dir, vector<string> outputs);
27
28  public:
29      vector<vector<pair<QString,QString>>> handle(QString dir); //返回两个列表, ans[0]是等价, ans[1]是不等价
30
31 };

```

b) 具体函数实现如图 (实验四报告) :

四、代码设计

(GUI 部分在实验五中实现, 目前实现的是命令行交互形式)

- 运行流程

```

graph LR
    Start([START]) --> Step1[输入待处理的程序语句]
    Step1 --> Step2[初识子目录]
    Step2 --> Step3[遍历每个子目录  
和递归入  
1. 读取输入文件  
2. 读取重定向或定义文件  
3. 脚本语言不可执行, 跳转到分类]
    Step3 --> Finish([FINISH])

```

- 核心函数
 - (1) void handle(string path): 接收输入, 调用(2), (3), (4)中的函数, 控制整个逻辑
 - 函数最初会先清空上次执行过后的遗留结果文件夹 (如果有的话)
 - 后创建本次运行后续要用的子目录。

在上述代码中, 通过 system()函数来调用 shell 命令 mkdir 和 rm -rf.

(2) 处理文件函数:

- vector<string> getDirList(string dirpath): 获取输入目录下的所有文件夹
- vector<string> getDirList(string dirpath): 获取子目录下的所有文件

```

12 //vector<string> getDirList(string dirpath){//读取input下的所有文件夹
13 {
14     vector<string> allDir;
15     DIR *dir = opendir(dirpath.c_str());
16     if (dir == NULL)
17     {
18         cout << "openDir error" << endl;
19         return allDir;
20     }
21     struct dirent *entry;
22     while ((entry = readdir(dir)) != NULL)
23     {
24         if (entry->type == DT_DIR){//读取子目录
25             string g = dirpath + '/' + entry->d_name;
26             allDir.push_back(g);
27             if (entry->d_type == DT_REG){//读取子文件
28                 string filename = getlist(entryName); //设置有子进程
29                 allDir.push_back(filename);
30             }
31         }
32     }
33     closedir(dir);
34 }
35 //vector<string> getDirList(string dirpath){//读取所有文件的路径
36 {
37     vector<string> allFile;
38     for (int i=0; i<allDir.size(); i++)
39     {
40         string filename = allDir[i].c_str();
41         cout << filename << endl;
42     }
43 }
44 }

```

8

9

```

40 //vector<string> getDirList(string dirpath){//读取input下的所有文件夹
41 {
42     vector<string> allDir;
43     DIR *dir = opendir(dirpath.c_str());
44     if (dir == NULL)
45     {
46         cout << "openDir error" << endl;
47         return allDir;
48     }
49     struct dirent *entry;
50     while ((entry = readdir(dir)) != NULL)
51     {
52         if (entry->type == DT_DIR){//读取子目录
53             string g = dirpath + '/' + entry->d_name;
54             allDir.push_back(g);
55             if (entry->d_type == DT_REG){//读取子文件
56                 string filename = getlist(entryName); //设置有子进程
57                 allDir.push_back(filename);
58             }
59         }
60     }
61     closedir(dir);
62 }
63 //vector<string> getDirList(string dirpath){//读取所有文件的路径
64 {
65     vector<string> allFile;
66     for (int i=0; i<allDir.size(); i++)
67     {
68         string filename = allDir[i].c_str();
69         cout << filename << endl;
70     }
71 }
72 }

```

在上述代码中, 通过使用 DIR, dirent 数据结构和 opendir 函数, 实现了对目录或目录中的文件进行操作的目的。结构体 dirent 中的成员 d_type 表示读取的目录中的文件的类型, d_name 表示读取的目录中的文件的名字。函数 DIR opendir(const char *pathname), 即打开文件目录, 返回的就是指向 DIR 结构体的指针。

(3) string inputPara(string dir, string txtPath): 根据标准输入文件在子目录下创建随机输入, 同时返回随机输入文件路径。

```

73 size_t pos1 = temp.find('C');//分割
74 size_t pos2=temp.find('C');//获得类型
75 size_t pos3=temp.find('C');//获得类型
76 string type=temp.substr(0, pos1); //获得类型
77 int x=std::stoi(temp.substr(pos1, pos2-pos1-1));
78 int y=std::stoi(temp.substr(pos2+1, pos3-pos2-1));
79
80 if(type=="int")
81 {
82     string tmp="";
83     int len=(rand() % (y-x+1))+x;//随机长度
84     for(int i=0; i<len; ++i)
85     {
86         tmp+= 'a' + rand() % 26; //随机字符
87     }
88     fout<<tmp<<" ";
89 }

```

如图所示, 在本函数中主要使用字符串切割和随机函数 rand() 来解析 stdin_formatted.txt 文件, 构造随机输入。

(4) string exec(string dir, string input, string cpp): 执行 cpp 文件, 并将输出文件保存至 output 目录下

```

183 DIR *dir;
184 cout<< cpp << endl;
185
186 //编译输出文件路径名
187 size_t pos=cpp.find(".");
188 string outPath=cpp.substr(0, pos)<<".output.txt"; //file.cpp编译后为output.txt
189
190 //file.c, file.cpp, find last of ".c", pos+pos-1); //gcc生成的可执行文件
191 string cmd="gcc "+cpp+" -o "+outPath;
192 string execute(cpp.substr(0, pos)); //gcc生成的可执行文件
193
194 if(outPath == NULL)
195 {
196     //string cmd1="grep -l 'yourput'" // xargs rm -rf &;
197     //string cmd2="cd "+dir;
198     //string cmd3="rm "+outPath;
199     //string cmd4="cp "+outPath+" -o "+execFile+" "+cpp;
200     //string cmd5="gcc "+cpp+" -o "+execFile;
201     //string cmd6="rm "+outPath;
202     string cmd2=execFile + <> " <> "+outPath;
203     cout<< cmd2 << endl << cmd1 << endl;
204
205 system("char>cmd1 << endl");
206 system("char>cmd2 << endl");
207
208 //popen("char>cmd1 << endl", "r"); //p指向缓冲区 (存疑)
209 //if (p != NULL)
210 //    pclose(p);
211 j = cout << " fail to get command";
212
213 //for (int i=0; i<outPath.length(); i++)
214 //    cout << outPath[i];
215
216 //for (int i=0; i<outPath.length(); i++)
217 //    cout << outPath[i];
218
219 //cout << endl;
220
221 //cout << endl;
222
223 //cout << endl;
224
225 //cout << endl;
226
227 //cout << endl;
228
229 //cout << endl;
230
231 //cout << endl;
232
233 //cout << endl;
234
235 //cout << endl;
236
237 //cout << endl;
238
239 //cout << endl;
240
241 //cout << endl;
242
243 //cout << endl;
244
245 //cout << endl;
246
247 //cout << endl;
248
249 //cout << endl;
250
251 //cout << endl;
252
253 //cout << endl;
254
255 //cout << endl;
256
257 //cout << endl;
258
259 //cout << endl;
260
261 //cout << endl;
262
263 //cout << endl;
264
265 //cout << endl;
266
267 //cout << endl;
268
269 //cout << endl;
270
271 //cout << endl;
272
273 //cout << endl;
274
275 //cout << endl;
276
277 //cout << endl;
278
279 //cout << endl;
280
281 //cout << endl;
282
283 //cout << endl;
284
285 //cout << endl;
286
287 //cout << endl;
288
289 //cout << endl;
290
291 //cout << endl;
292
293 //cout << endl;
294
295 //cout << endl;
296
297 //cout << endl;
298
299 //cout << endl;
300
301 //cout << endl;
302
303 //cout << endl;
304
305 //cout << endl;
306
307 //cout << endl;
308
309 //cout << endl;
310
311 //cout << endl;
312
313 //cout << endl;
314
315 //cout << endl;
316
317 //cout << endl;
318
319 //cout << endl;
320
321 //cout << endl;
322
323 //cout << endl;
324
325 //cout << endl;
326
327 //cout << endl;
328
329 //cout << endl;
330
331 //cout << endl;
332
333 //cout << endl;
334
335 //cout << endl;
336
337 //cout << endl;
338
339 //cout << endl;
340
341 //cout << endl;
342
343 //cout << endl;
344
345 //cout << endl;
346
347 //cout << endl;
348
349 //cout << endl;
350
351 //cout << endl;
352
353 //cout << endl;
354
355 //cout << endl;
356
357 //cout << endl;
358
359 //cout << endl;
360
361 //cout << endl;
362
363 //cout << endl;
364
365 //cout << endl;
366
367 //cout << endl;
368
369 //cout << endl;
370
371 //cout << endl;
372
373 //cout << endl;
374
375 //cout << endl;
376
377 //cout << endl;
378
379 //cout << endl;
380
381 //cout << endl;
382
383 //cout << endl;
384
385 //cout << endl;
386
387 //cout << endl;
388
389 //cout << endl;
390
391 //cout << endl;
392
393 //cout << endl;
394
395 //cout << endl;
396
397 //cout << endl;
398
399 //cout << endl;
400
401 //cout << endl;
402
403 //cout << endl;
404
405 //cout << endl;
406
407 //cout << endl;
408
409 //cout << endl;
410
411 //cout << endl;
412
413 //cout << endl;
414
415 //cout << endl;
416
417 //cout << endl;
418
419 //cout << endl;
420
421 //cout << endl;
422
423 //cout << endl;
424
425 //cout << endl;
426
427 //cout << endl;
428
429 //cout << endl;
430
431 //cout << endl;
432
433 //cout << endl;
434
435 //cout << endl;
436
437 //cout << endl;
438
439 //cout << endl;
440
441 //cout << endl;
442
443 //cout << endl;
444
445 //cout << endl;
446
447 //cout << endl;
448
449 //cout << endl;
450
451 //cout << endl;
452
453 //cout << endl;
454
455 //cout << endl;
456
457 //cout << endl;
458
459 //cout << endl;
460
461 //cout << endl;
462
463 //cout << endl;
464
465 //cout << endl;
466
467 //cout << endl;
468
469 //cout << endl;
470
471 //cout << endl;
472
473 //cout << endl;
474
475 //cout << endl;
476
477 //cout << endl;
478
479 //cout << endl;
480
481 //cout << endl;
482
483 //cout << endl;
484
485 //cout << endl;
486
487 //cout << endl;
488
489 //cout << endl;
490
491 //cout << endl;
492
493 //cout << endl;
494
495 //cout << endl;
496
497 //cout << endl;
498
499 //cout << endl;
500
501 //cout << endl;
502
503 //cout << endl;
504
505 //cout << endl;
506
507 //cout << endl;
508
509 //cout << endl;
510
511 //cout << endl;
512
513 //cout << endl;
514
515 //cout << endl;
516
517 //cout << endl;
518
519 //cout << endl;
520
521 //cout << endl;
522
523 //cout << endl;
524
525 //cout << endl;
526
527 //cout << endl;
528
529 //cout << endl;
530
531 //cout << endl;
532
533 //cout << endl;
534
535 //cout << endl;
536
537 //cout << endl;
538
539 //cout << endl;
540
541 //cout << endl;
542
543 //cout << endl;
544
545 //cout << endl;
546
547 //cout << endl;
548
549 //cout << endl;
550
551 //cout << endl;
552
553 //cout << endl;
554
555 //cout << endl;
556
557 //cout << endl;
558
559 //cout << endl;
560
561 //cout << endl;
562
563 //cout << endl;
564
565 //cout << endl;
566
567 //cout << endl;
568
569 //cout << endl;
570
571 //cout << endl;
572
573 //cout << endl;
574
575 //cout << endl;
576
577 //cout << endl;
578
579 //cout << endl;
580
581 //cout << endl;
582
583 //cout << endl;
584
585 //cout << endl;
586
587 //cout << endl;
588
589 //cout << endl;
590
591 //cout << endl;
592
593 //cout << endl;
594
595 //cout << endl;
596
597 //cout << endl;
598
599 //cout << endl;
600
601 //cout << endl;
602
603 //cout << endl;
604
605 //cout << endl;
606
607 //cout << endl;
608
609 //cout << endl;
610
611 //cout << endl;
612
613 //cout << endl;
614
615 //cout << endl;
616
617 //cout << endl;
618
619 //cout << endl;
620
621 //cout << endl;
622
623 //cout << endl;
624
625 //cout << endl;
626
627 //cout << endl;
628
629 //cout << endl;
630
631 //cout << endl;
632
633 //cout << endl;
634
635 //cout << endl;
636
637 //cout << endl;
638
639 //cout << endl;
640
641 //cout << endl;
642
643 //cout << endl;
644
645 //cout << endl;
646
647 //cout << endl;
648
649 //cout << endl;
650
651 //cout << endl;
652
653 //cout << endl;
654
655 //cout << endl;
656
657 //cout << endl;
658
659 //cout << endl;
660
661 //cout << endl;
662
663 //cout << endl;
664
665 //cout << endl;
666
667 //cout << endl;
668
669 //cout << endl;
670
671 //cout << endl;
672
673 //cout << endl;
674
675 //cout << endl;
676
677 //cout << endl;
678
679 //cout << endl;
680
681 //cout << endl;
682
683 //cout << endl;
684
685 //cout << endl;
686
687 //cout << endl;
688
689 //cout << endl;
690
691 //cout << endl;
692
693 //cout << endl;
694
695 //cout << endl;
696
697 //cout << endl;
698
699 //cout << endl;
700
701 //cout << endl;
702
703 //cout << endl;
704
705 //cout << endl;
706
707 //cout << endl;
708
709 //cout << endl;
710
711 //cout << endl;
712
713 //cout << endl;
714
715 //cout << endl;
716
717 //cout << endl;
718
719 //cout << endl;
720
721 //cout << endl;
722
723 //cout << endl;
724
725 //cout << endl;
726
727 //cout << endl;
728
729 //cout << endl;
730
731 //cout << endl;
732
733 //cout << endl;
734
735 //cout << endl;
736
737 //cout << endl;
738
739 //cout << endl;
740
741 //cout << endl;
742
743 //cout << endl;
744
745 //cout << endl;
746
747 //cout << endl;
748
749 //cout << endl;
750
751 //cout << endl;
752
753 //cout << endl;
754
755 //cout << endl;
756
757 //cout << endl;
758
759 //cout << endl;
760
761 //cout << endl;
762
763 //cout << endl;
764
765 //cout << endl;
766
767 //cout << endl;
768
769 //cout << endl;
770
771 //cout << endl;
772
773 //cout << endl;
774
775 //cout << endl;
776
777 //cout << endl;
778
779 //cout << endl;
780
781 //cout << endl;
782
783 //cout << endl;
784
785 //cout << endl;
786
787 //cout << endl;
788
789 //cout << endl;
790
791 //cout << endl;
792
793 //cout << endl;
794
795 //cout << endl;
796
797 //cout << endl;
798
799 //cout << endl;
800
801 //cout << endl;
802
803 //cout << endl;
804
805 //cout << endl;
806
807 //cout << endl;
808
809 //cout << endl;
810
811 //cout << endl;
812
813 //cout << endl;
814
815 //cout << endl;
816
817 //cout << endl;
818
819 //cout << endl;
820
821 //cout << endl;
822
823 //cout << endl;
824
825 //cout << endl;
826
827 //cout << endl;
828
829 //cout << endl;
830
831 //cout << endl;
832
833 //cout << endl;
834
835 //cout << endl;
836
837 //cout << endl;
838
839 //cout << endl;
840
841 //cout << endl;
842
843 //cout << endl;
844
845 //cout << endl;
846
847 //cout << endl;
848
849 //cout << endl;
850
851 //cout << endl;
852
853 //cout << endl;
854
855 //cout << endl;
856
857 //cout << endl;
858
859 //cout << endl;
860
861 //cout << endl;
862
863 //cout << endl;
864
865 //cout << endl;
866
867 //cout << endl;
868
869 //cout << endl;
870
871 //cout << endl;
872
873 //cout << endl;
874
875 //cout << endl;
876
877 //cout << endl;
878
879 //cout << endl;
880
881 //cout << endl;
882
883 //cout << endl;
884
885 //cout << endl;
886
887 //cout << endl;
888
889 //cout << endl;
890
891 //cout << endl;
892
893 //cout << endl;
894
895 //cout << endl;
896
897 //cout << endl;
898
899 //cout << endl;
900
901 //cout << endl;
902
903 //cout << endl;
904
905 //cout << endl;
906
907 //cout << endl;
908
909 //cout << endl;
910
911 //cout << endl;
912
913 //cout << endl;
914
915 //cout << endl;
916
917 //cout << endl;
918
919 //cout << endl;
920
921 //cout << endl;
922
923 //cout << endl;
924
925 //cout << endl;
926
927 //cout << endl;
928
929 //cout << endl;
930
931 //cout << endl;
932
933 //cout << endl;
934
935 //cout << endl;
936
937 //cout << endl;
938
939 //cout << endl;
940
941 //cout << endl;
942
943 //cout << endl;
944
945 //cout << endl;
946
947 //cout << endl;
948
949 //cout << endl;
950
951 //cout << endl;
952
953 //cout << endl;
954
955 //cout << endl;
956
957 //cout << endl;
958
959 //cout << endl;
960
961 //cout << endl;
962
963 //cout << endl;
964
965 //cout << endl;
966
967 //cout << endl;
968
969 //cout << endl;
970
971 //cout << endl;
972
973 //cout << endl;
974
975 //cout << endl;
976
977 //cout << endl;
978
979 //cout << endl;
980
981 //cout << endl;
982
983 //cout << endl;
984
985 //cout << endl;
986
987 //cout << endl;
988
989 //cout << endl;
990
991 //cout << endl;
992
993 //cout << endl;
994
995 //cout << endl;
996
997 //cout << endl;
998
999 //cout << endl;
1000
1001 //cout << endl;
1002
1003 //cout << endl;
1004
1005 //cout << endl;
1006
1007 //cout << endl;
1008
1009 //cout << endl;
1010
1011 //cout << endl;
1012
1013 //cout << endl;
1014
1015 //cout << endl;
1016
1017 //cout << endl;
1018
1019 //cout << endl;
1020
1021 //cout << endl;
1022
1023 //cout << endl;
1024
1025 //cout << endl;
1026
1027 //cout << endl;
1028
1029 //cout << endl;
1030
1031 //cout << endl;
1032
1033 //cout << endl;
1034
1035 //cout << endl;
1036
1037 //cout << endl;
1038
1039 //cout << endl;
1040
1041 //cout << endl;
1042
1043 //cout << endl;
1044
1045 //cout << endl;
1046
1047 //cout << endl;
1048
1049 //cout << endl;
1050
1051 //cout << endl;
1052
1053 //cout << endl;
1054
1055 //cout << endl;
1056
1057 //cout << endl;
1058
1059 //cout << endl;
1060
1061 //cout << endl;
1062
1063 //cout << endl;
1064
1065 //cout << endl;
1066
1067 //cout << endl;
1068
1069 //cout << endl;
1070
1071 //cout << endl;
1072
1073 //cout << endl;
1074
1075 //cout << endl;
1076
1077 //cout << endl;
1078
1079 //cout << endl;
1080
1081 //cout << endl;
1082
1083 //cout << endl;
1084
1085 //cout << endl;
1086
1087 //cout << endl;
1088
1089 //cout << endl;
1090
1091 //cout << endl;
1092
1093 //cout << endl;
1094
1095 //cout << endl;
1096
1097 //cout << endl;
1098
1099 //cout << endl;
1100
1101 //cout << endl;
1102
1103 //cout << endl;
1104
1105 //cout << endl;
1106
1107 //cout << endl;
1108
1109 //cout << endl;
1110
1111 //cout << endl;
1112
1113 //cout << endl;
1114
1115 //cout << endl;
1116
1117 //cout << endl;
1118
1119 //cout << endl;
1120
1121 //cout << endl;
1122
1123 //cout << endl;
1124
1125 //cout << endl;
1126
1127 //cout << endl;
1128
1129 //cout << endl;
1130
1131 //cout << endl;
1132
1133 //cout << endl;
1134
1135 //cout << endl;
1136
1137 //cout << endl;
1138
1139 //cout << endl;
1140
1141 //cout << endl;
1142
1143 //cout << endl;
1144
1145 //cout << endl;
1146
1147 //cout << endl;
1148
1149 //cout << endl;
1150
1151 //cout << endl;
1152
1153 //cout << endl;
1154
1155 //cout << endl;
1156
1157 //cout << endl;
1158
1159 //cout << endl;
1160
1161 //cout << endl;
1162
1163 //cout << endl;
1164
1165 //cout << endl;
1166
1167 //cout << endl;
1168
1169 //cout << endl;
1170
1171 //cout << endl;
1172
1173 //cout << endl;
1174
1175 //cout << endl;
1176
1177 //cout << endl;
1178
1179 //cout << endl;
1180
1181 //cout << endl;
1182
1183 //cout << endl;
1184
1185 //cout << endl;
1186
1187 //cout << endl;
1188
1189 //cout << endl;
1190
1191 //cout << endl;
1192
1193 //cout << endl;
1194
1195 //cout << endl;
1196
1197 //cout << endl;
1198
1199 //cout << endl;
1200
1201 //cout << endl;
1202
1203 //cout << endl;
1204
1205 //cout << endl;
1206
1207 //cout << endl;
1208
1209 //cout << endl;
1210
1211 //cout << endl;
1212
1213 //cout << endl;
1214
1215 //cout << endl;
1216
1217 //cout << endl;
1218
1219 //cout << endl;
1220
1221 //cout << endl;
1222
1223 //cout << endl;
1224
1225 //cout << endl;
1226
1227 //cout << endl;
1228
1229 //cout << endl;
1230
1231 //cout << endl;
1232
1233 //cout << endl;
1234
1235 //cout << endl;
1236
1237 //cout << endl;
1238
1239 //cout << endl;
1240
1241 //cout << endl;
1242
1243 //cout << endl;
1244
1245 //cout << endl;
1246
1247 //cout << endl;
1248
1249 //cout << endl;
1250
1251 //cout << endl;
1252
1253 //cout << endl;
1254
1255 //cout << endl;
1256
1257 //cout << endl;
1258
1259 //cout << endl;
1260
1261 //cout << endl;
1262
1263 //cout << endl;
1264
1265 //cout << endl;
1266
1267 //cout << endl;
1268
1269 //cout << endl;
1270
1271 //cout << endl;
1272
1273 //cout << endl;
1274
1275 //cout << endl;
1276
1277 //cout << endl;
1278
1279 //cout << endl;
1280
1281 //cout << endl;
1282
1283 //cout << endl;
1284
1285 //cout << endl;
1286
1287 //cout << endl;
1288
1289 //cout << endl;
1290
1291 //cout << endl;
1292
1293 //cout << endl;
1294
1295 //cout << endl;
1296
1297 //cout << endl;
1298
1299 //cout << endl;
1300
1301 //cout << endl;
1302
1303
```

```
vector<vector<pair<QString,QString>>> ans=Handle().handle(dir);
```

3. GUI 模块交互控制代码

- a. 初始化窗口，进行一定的图片渲染，设计用户友好的界面

```
13  MainWindow::MainWindow(QWidget *parent)
14      : QMainWindow(parent)
15  , ui(new Ui::MainWindow)
16  {
17      ui->setupUi(this);
18  }
19
20  ~MainWindow()
21  {
22      delete ui;          重写QMainwindow::paintEvent函数
23 }                                来自自定义绘制窗口
24
25 void MainWindow::paintEvent(QPaintEvent *event)
26 {
27     QPainter painter(this);
28     painter.drawPixmap(rect(),QPixmap(":/image/bkg.jpg"),QRect());
29 }
30
31
32 }
```

- b. 通过读取窗口中文本编辑框的内容来获得用户输入的目录路径并执行

```
35 void MainWindow::on_pushButton_clicked()//开始检测按钮
36 {
37     dir=ui->lineEdit->text(); 读取用户输入
38     if(dir==NULL)
39     {
40         QMessageBox::information(this, "warning","The directory path can not be NULL!");
41         return;
42     }
43
44 //string path="/Users/pro/Desktop/QtProjects/SoftwareProj/input";
45 //vector<vector<pair<QString,QString>>> ans=Handle().handle(QString::fromStdString(path));
46 vector<vector<pair<QString,QString>>> ans=Handle().handle(dir);
47 equal=ans[0];
48 unequal=ans[1];
49
50 showAns();
51
52 }
```

- c. 使用 **Qt::TableView** 控件，将结果输出至 **Qt::TableView** 中，实现类似表格的功能，同时对表格进行渲染，实现用户友好的界面

```

54 void MainWindow::showAns()
55 {
56     QStandardItemModel* model = new QStandardItemModel(); 实现表格
57     QStringList labels = QOBJECT::trUtf8("file1,file2").simplified().split(",");
58     model->setHorizontalHeaderLabels(labels);
59
60     QStandardItem* item;
61     int i=0;
62     while(i!=equal.size()) △comparison of integers of different signs: 'int' and 'std::__1::vector<st
63     {
64         item = new QStandardItem(equal[i].first); //括号里面是QString即可
65         model->setItem(i,0,item);
66         item = new QStandardItem(equal[i].second);
67         model->setItem(i,1,item);
68         i++;
69     }
70
71     ui->tableView->setModel(model);
72     ui->tableView->setColumnWidth(0, 500); //设置列宽, 注意要在setModel之后
73     ui->tableView->setColumnWidth(1, 500);
74     ui->tableView->show();
75     ui->tableView->setEditTriggers(QAbstractItemView::NoEditTriggers);
76     ui->label_2->setText("Equal Result: total "+QString::number(equal.size(), 10)+".");
77     ui->label_2->adjustSize();
78     ui->label_2->repaint();
79
80     QStandardItemModel* model2 = new QStandardItemModel();
81     QStringList labels2 = QOBJECT::trUtf8("File1,File2").simplified().split(",");
82     model2->setHorizontalHeaderLabels(labels2); △'trUtf8' is deprecated
83
84     //QStandardItem* item;
85     i=0;

```



```

54 void MainWindow::showAns()
55 {
56     QStandardItemModel* model = new QStandardItemModel(); 实现表格
57     QStringList labels = QOBJECT::trUtf8("file1,file2").simplified().split(",");
58     model->setHorizontalHeaderLabels(labels);
59
60     QStandardItem* item;
61     int i=0;
62     while(i!=equal.size()) △comparison of integers of different signs: 'int' and 'std::__1::vector<st
63     {
64         item = new QStandardItem(equal[i].first); //括号里面是QString即可
65         model->setItem(i,0,item);
66         item = new QStandardItem(equal[i].second);
67         model->setItem(i,1,item);
68         i++;
69     }
70
71     ui->tableView->setModel(model);
72     ui->tableView->setColumnWidth(0, 500); //设置列宽, 注意要在setModel之后
73     ui->tableView->setColumnWidth(1, 500);
74     ui->tableView->show();
75     ui->tableView->setEditTriggers(QAbstractItemView::NoEditTriggers);
76     ui->label_2->setText("Equal Result: total "+QString::number(equal.size(), 10)+".");
77     ui->label_2->adjustSize();
78     ui->label_2->repaint();
79
80     QStandardItemModel* model2 = new QStandardItemModel();
81     QStringList labels2 = QOBJECT::trUtf8("File1,File2").simplified().split(",");
82     model2->setHorizontalHeaderLabels(labels2); △'trUtf8' is deprecated
83
84     //QStandardItem* item;
85     i=0;

```

d. 为窗口中的控件设置**槽函数**获取用户的点击事件，完成逻辑转换

```

105 void MainWindow::on_pushButton_2_clicked()//刷新, 数据改变后重新展示
106 {
107     showAns();
108 }
109
110 void MainWindow::on_commandLinkButton_clicked()//人工检测 槽函数
111 {
112     if(unequal.size()==0&&equal.size()==0)
113     {
114         QMessageBox::information(this, "warning","Please judge automatically first!");
115         return;
116     }
117     Manual m(&unequal,&equal);
118     m.exec();
119
120     showAns();
121 }
122
123

```

```

126 void Manual::on_commandLinkButton_clicked()//返回按钮
127 {
128     this->close();
129 }
130
131 void Manual::on_pushButton_clicked()//nextPair
132 {
133     if(isSave==false)//未保存需要提示是否保存
134     {
135         QMessageBox::StandardButton result= QMessageBox::question(this,"question","Are you sure not to save?");
136         if (result == QMessageBox::Save)
137         {
138             save();
139         }
140     }
141     if(tmpDir=="Unequal")
142     {
143         unequalIndex++;
144         if(unequalIndex==(*unequal).size())//说明不等价集遍历结束, 遍历等价集
145         {
146             tmpDir="Equal";
147         }

```

e. 将代码文件输出至 `Qt::TextBrowser` 中, 实现类似代码编辑器的功能

```

215 void Manual::showTextDiff()//展示不同
216 {
217     //清空选中框
218     ui->equal->setCheckState(Qt::Unchecked);
219     ui->unequal->setCheckState(Qt::Unchecked);
220     ui->notSure->setCheckState(Qt::Unchecked);
221
222     if(tmpDir=="Unequal")//确定打开哪个文件
223     {
224
225         QFile file1((*unequal)[unequalIndex].first); //path是文件的路径
226         QFile file2((*unequal)[unequalIndex].second); //path是文件的路径
227         if(!file1.open(QIODevice::ReadOnly | QIODevice::Text))
228         {
229             qDebug()<<"文件未打开!";
230         }
231         if(!file2.open(QIODevice::ReadOnly | QIODevice::Text))
232         {
233             qDebug()<<"文件未打开!";
234         }
235
236         //开始逐行读取文件并判断
237         while(!file1.atEnd()&&!file2.atEnd())
238         {
239             QByteArray array1 = file1.readLine();
240             QByteArray array2 = file2.readLine();
241             QString str1(array1);
242             QString str2(array2);
243
244             //一行相同, 直接输出
245             if(str1==str2)
246             {

```

```

247             ui->textBrowser->append(str1);
248             ui->textBrowser_2->append(str2);
249         }
250
251         //不同，以str1为主对file2逐行对比直至找到相同的，或者找不到
252     else
253     {
254         bool flag=false;
255         QString data1;
256         QString data2;
257         //制造空行高亮效果
258         ui->textBrowser->append("<span style=\"background-color: rgba(255,193,37,0.3);\"><font color=");
259         str2= "<font color=\"#FF0000\">`" + str2 + "</font>";
260         ui->textBrowser_2->append(str2);输出事件
261         while(!flag&&file1.atEnd()&&file2.atEnd())
262         {
263             array2 = file2.readLine();
264             QString tmp(array2);
265             if(str1==tmp)
266             {
267                 ui->textBrowser->append(str1);
268                 ui->textBrowser_2->append(tmp);
269                 flag=true;
270             }
271             else
272             {
273                 ui->textBrowser->append("<span style=\"background-color: rgba(255,193,37,0.3);\"><font color=");
274                 tmp= "<font color=\"#FF0000\">>" + tmp + "</font>";
275                 ui->textBrowser_2->append(tmp);
276             }
277         }
278     }
279 }
```

4. 自动推荐算法

采用排列组合方式，优先推荐不等价结果集中的代码对，都确认结束后推荐等价结果集中的代码对，最后提示所有文件都确认结束。

```

142     if(tmpDir=="Unequal")
143     {
144         unequalIndex++;
145         if(unequalIndex==(*unequal).size())说明不等价集遍历结束，遍历等价集 △ comparison of integers of different signs: 'int'...
146         {
147             tmpDir="Equal";
148         }
149     }
150     else
151     {
152         equalIndex++;
153         if(equalIndex==(*equal).size())说明遍历结束 △ comparison of integers of different signs: 'int'...
154         {
155             QMessageBox::information(this, "information", "That's all!");
156         }
157     }
}
```

5. 显示两个代码文件的不同 (Diff 命令, 扩展)

- 分析需求：实现类似于 Linux 的 Diff 命令，需要展示两个特点：（1）代码缺少之处空行加阴影（2）代码不同之处字体标红。
- 实现逻辑：
 - 共有两个代码文件 file1 和 file2，以 txt 文件格式存储在当前路径中；
 - 同时逐行读取文件中的代码行，存储在字符串 str1 和 str2 中；
 - 若字符串相同，则无格式输出至 QTextBrowser 控件中；
 - 若字符串不同，则以 file1 中读取的代码行 str1 为主，继续向下逐行对比 file2 中的代码行，直至找到相同的代码行；或者找不到相同的代码行，最后将 str2 输出；
 - 比对时，file1 中添加空行，同时以 html 格式添加文字背景输出至 QTextBrowser，file2 中不相同的行以 html 格式标红输出至 QTextBrowser；
 - 当一个文件遍历完时，将另一个没有遍历完的文件全部标红输出。

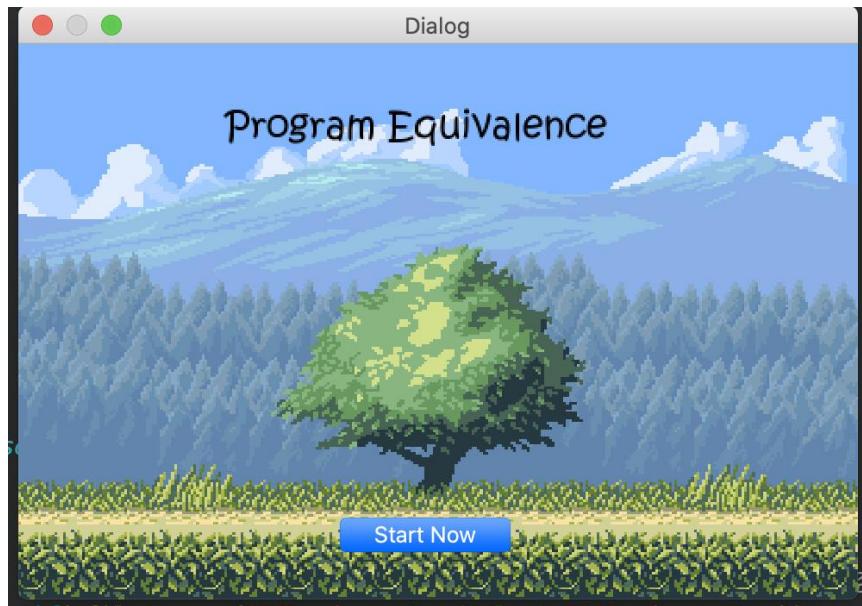
```

251 //不同，以str1为主对file2逐行对比直至找到相同的，或者找不到
252 ▼
253 {
254     bool flag=false;
255     QString data1;
256     QString data2;
257     //制造空行高亮效果
258     ui->textBrowser->append("<span style=\"background-color: rgba(255,193,37,0.3);\"><font color=\"#FF0000\">" + str2 + "</font> " ,利用html语法制造空行加阴影效果
259     str2= <font color="#FF0000"> + str2 + </font> ,
260     ui->textBrowser_2->append(str2);
261     while(!flag&&!file1.atEnd()&&!file2.atEnd())
262     {
263         array2 = file2.readLine();
264         QString tmp(array2);
265         if(str1==tmp)
266         {
267             ui->textBrowser->append(str1);
268             ui->textBrowser_2->append(tmp);
269             flag=true;
270         }
271     }
272     else
273     {
274         ui->textBrowser->append("<span style=\"background-color: rgba(255,193,37,0.3);\"><font color=\"#FF0000\">" + tmp + "</font> " ,利用html制造不同代码处加红效果
275         tmp= "<font color=\"#FF0000\">" + tmp + "</font>";
276         ui->textBrowser_2->append(tmp),
277     }
278 }
279 if(!flag)//没有找到，将str1输出
280 {
281     str1= "<font color=\"#FF0000\">" + str1 + "</font>";
282

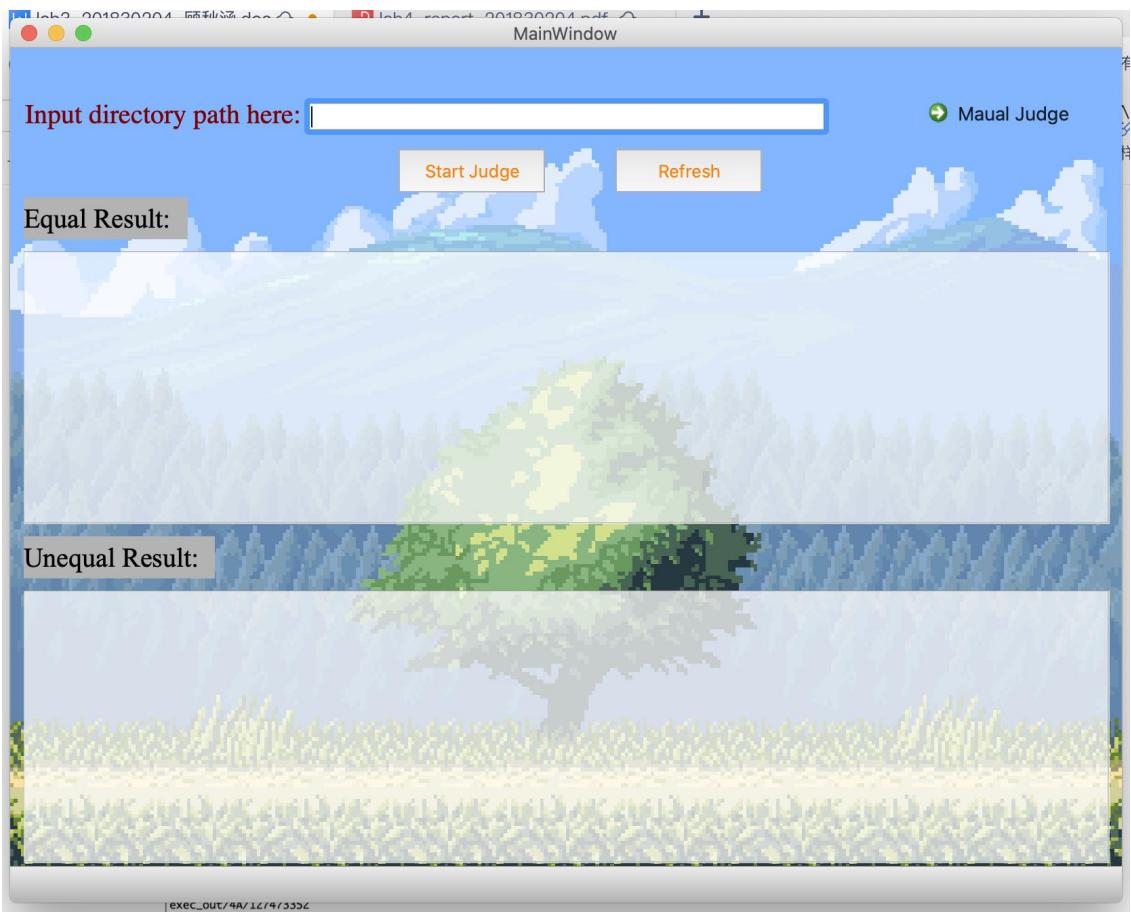
```

四、 实现展示

1. 开始界面



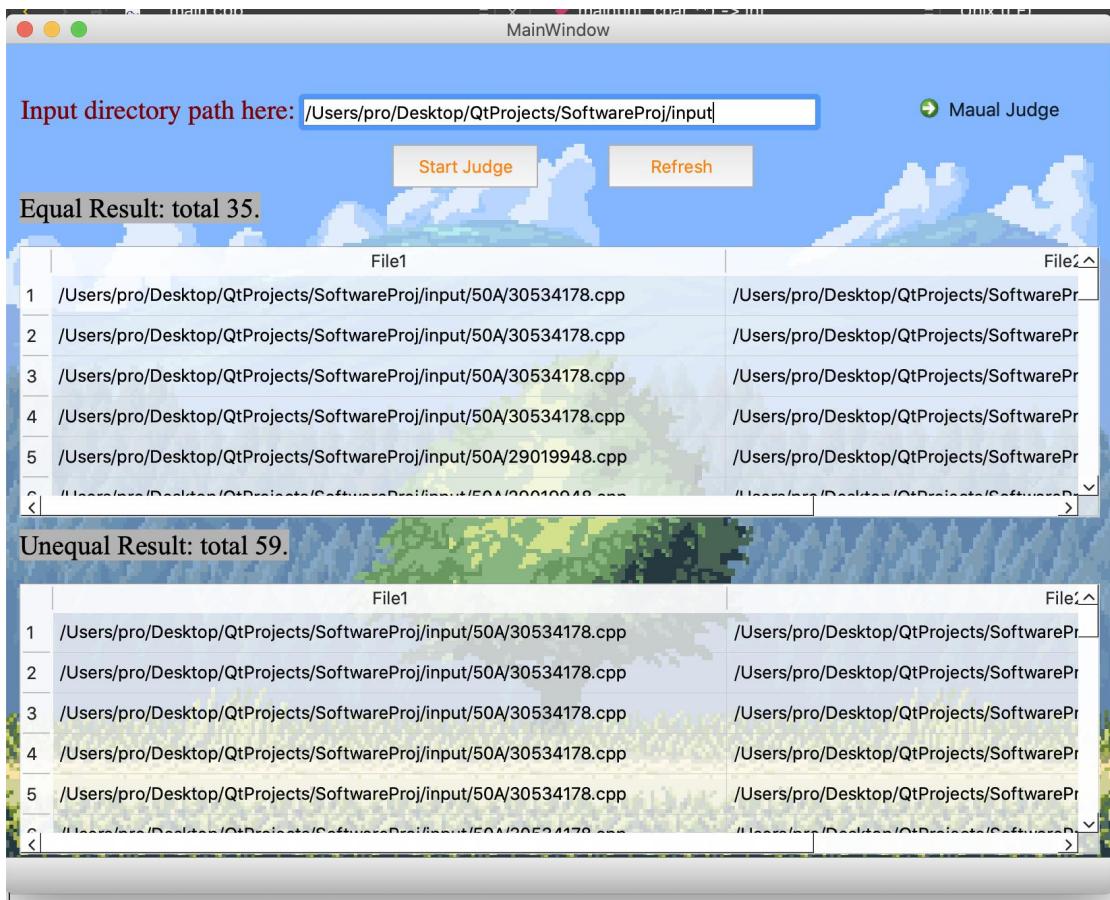
2. 进入后界面



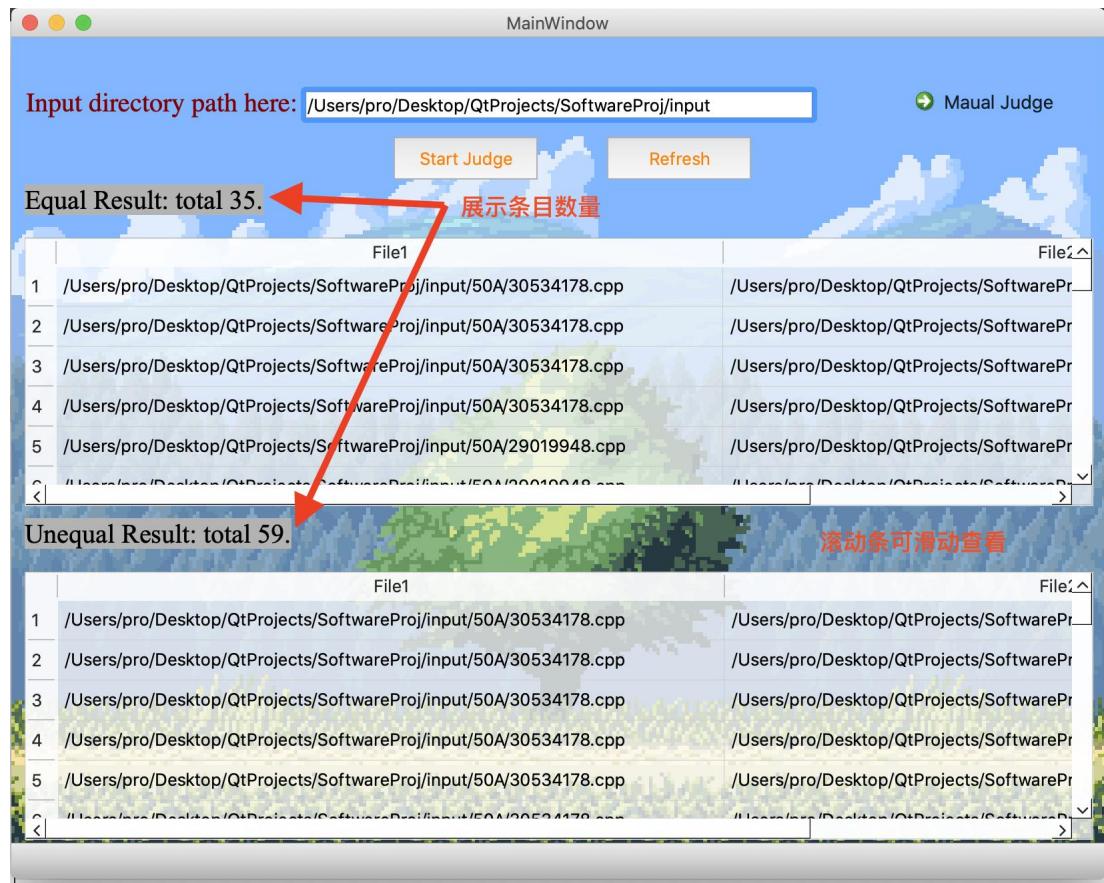
说明:



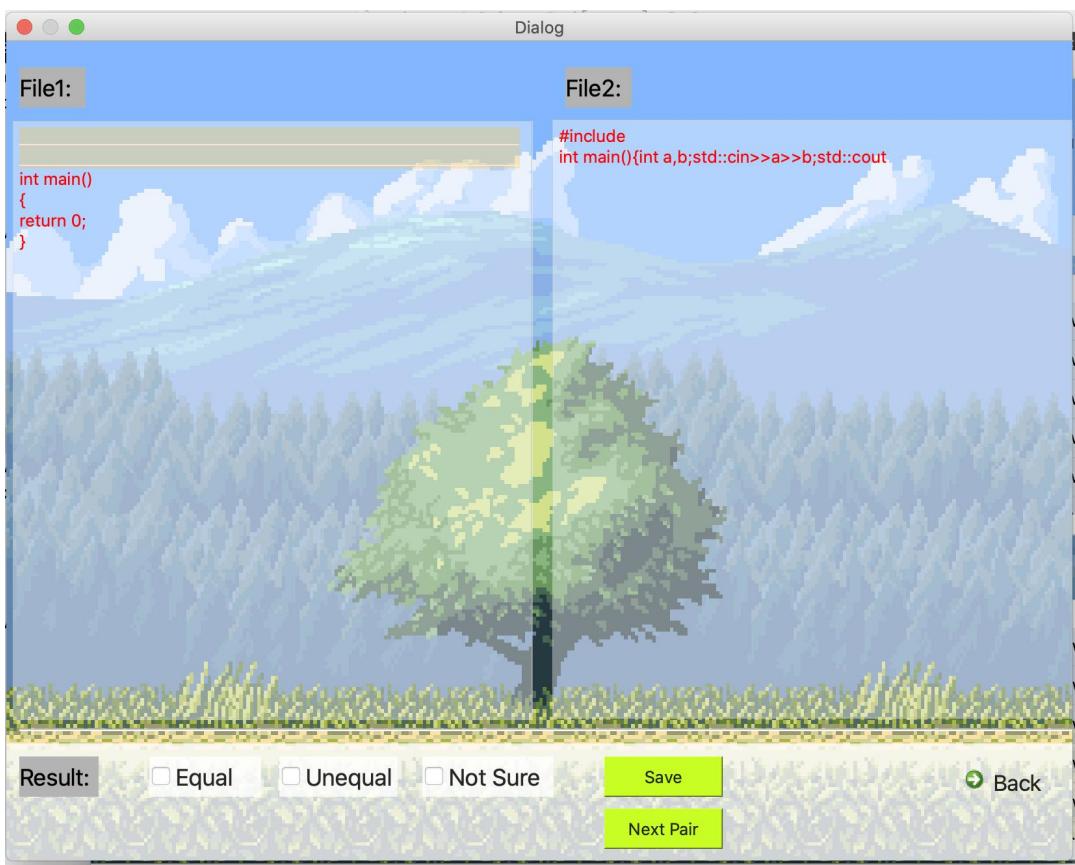
3. 程序展示结果



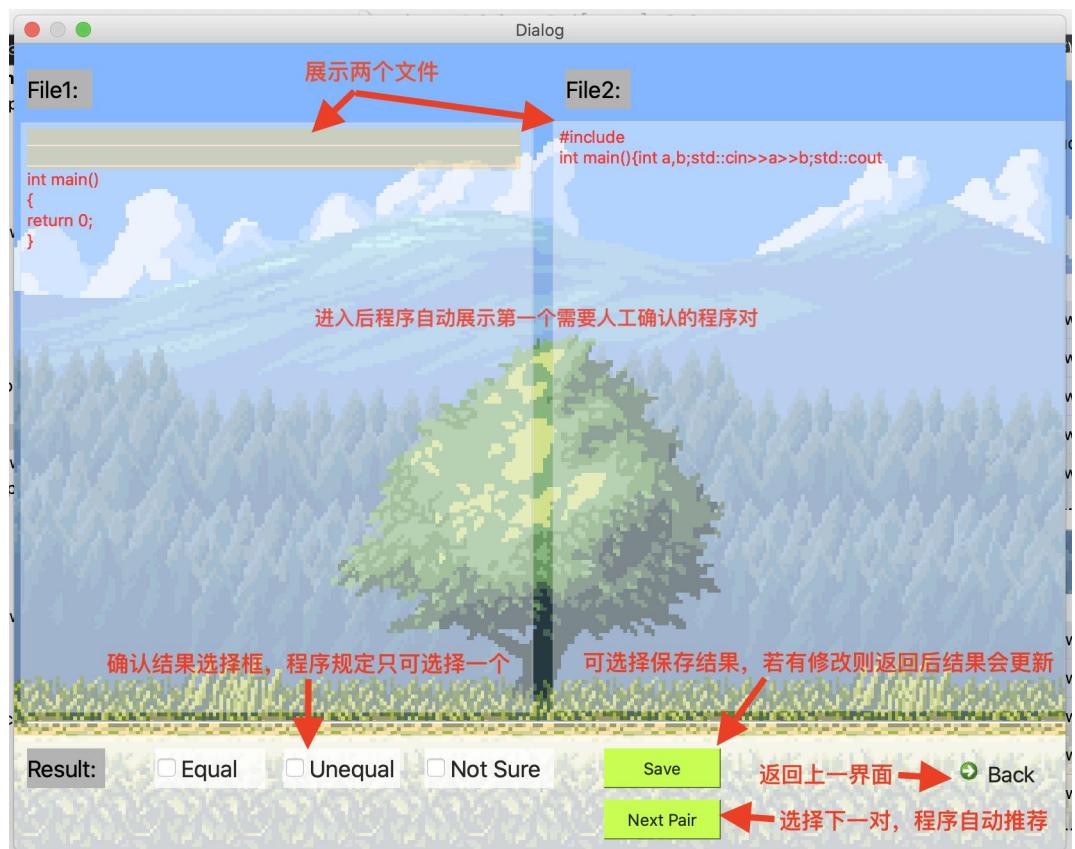
说明:



4. 点击进入人工检测

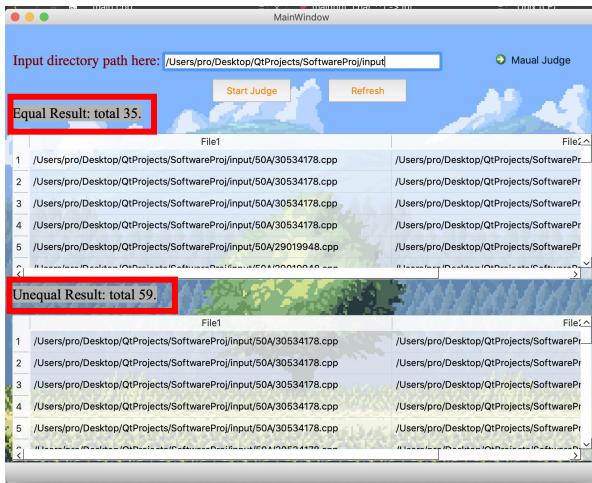


说明:

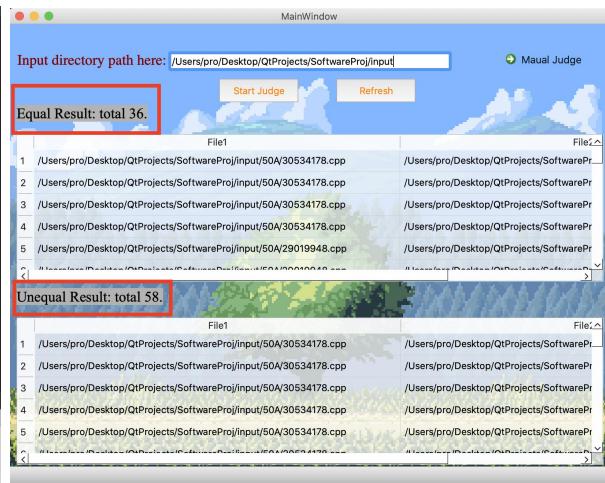


5. 人工检测返回后程序自动刷新更新结果

更新前:



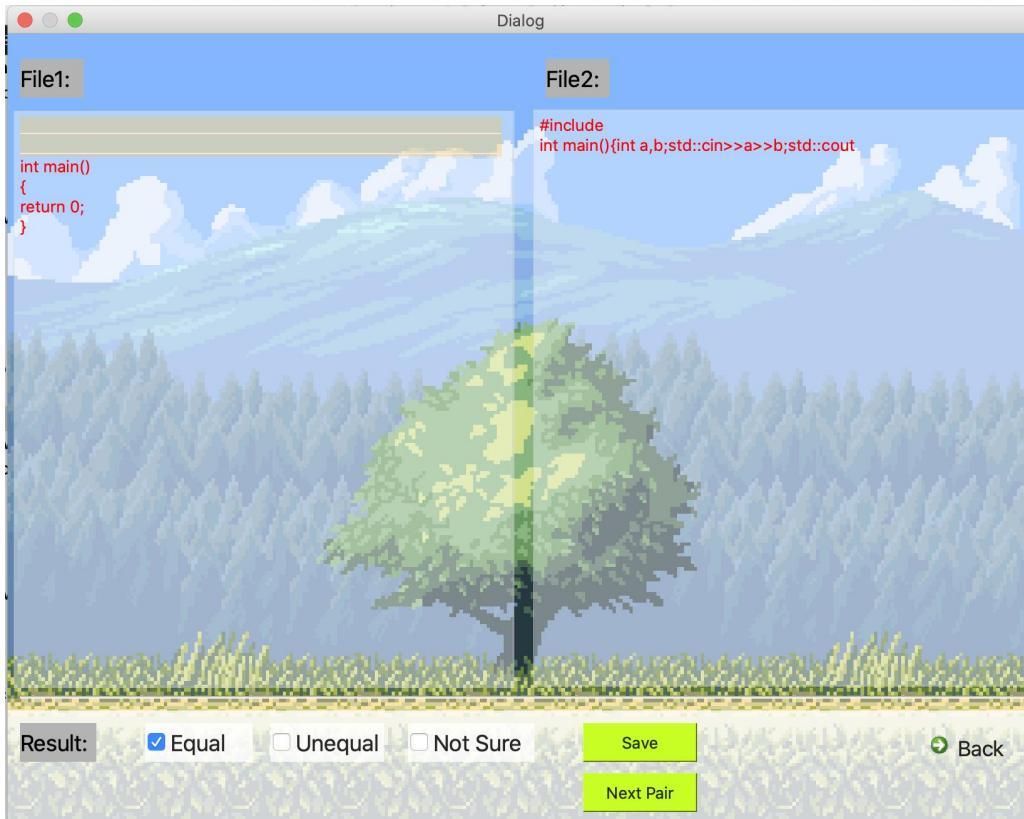
更新后:

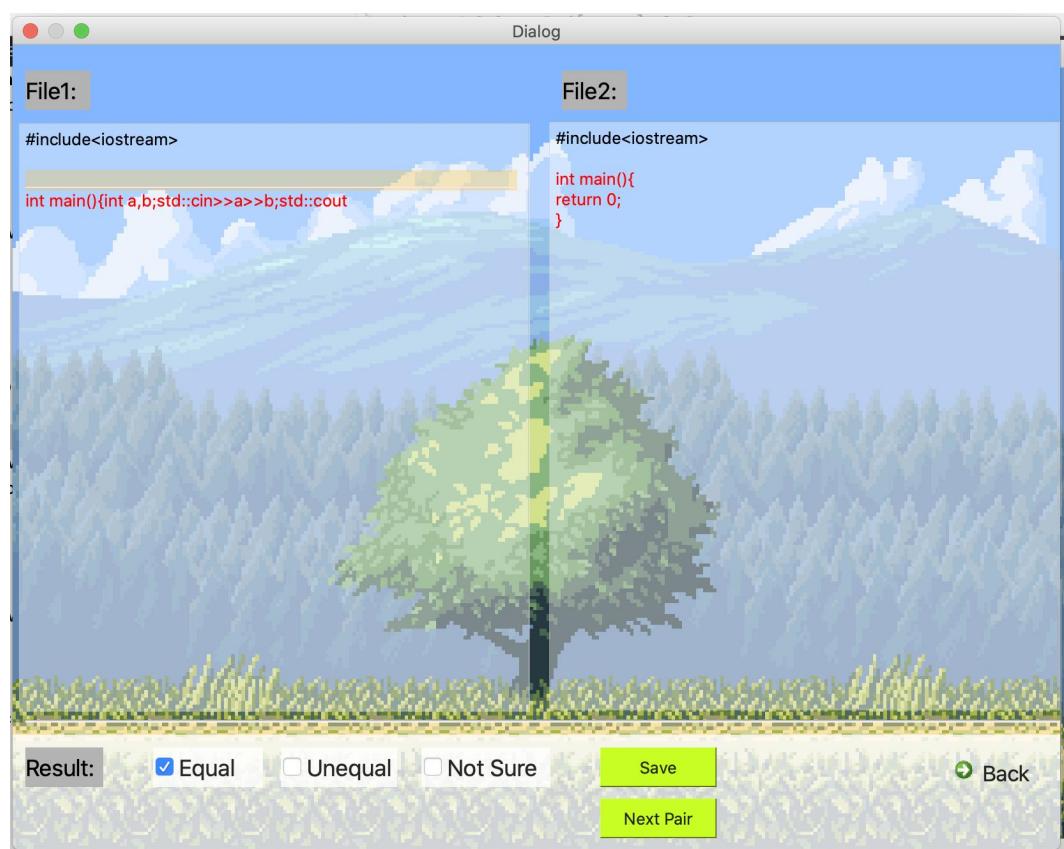
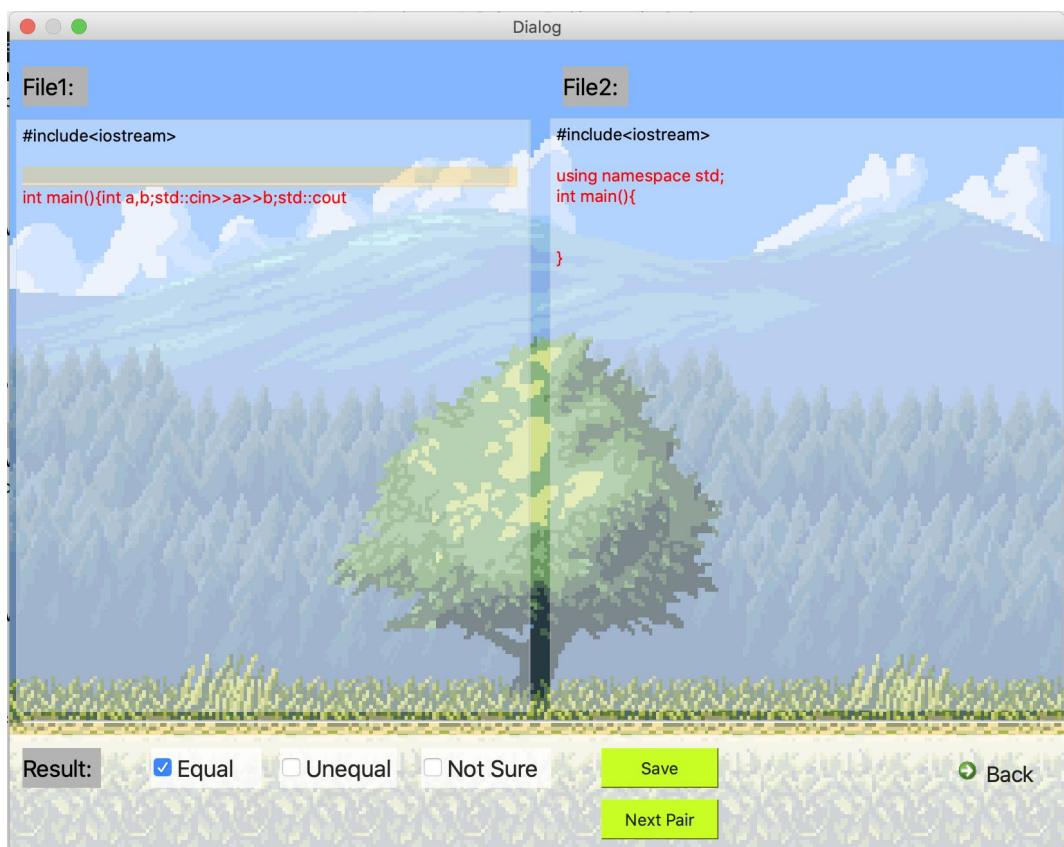


6. 左击左上角可关闭程序。

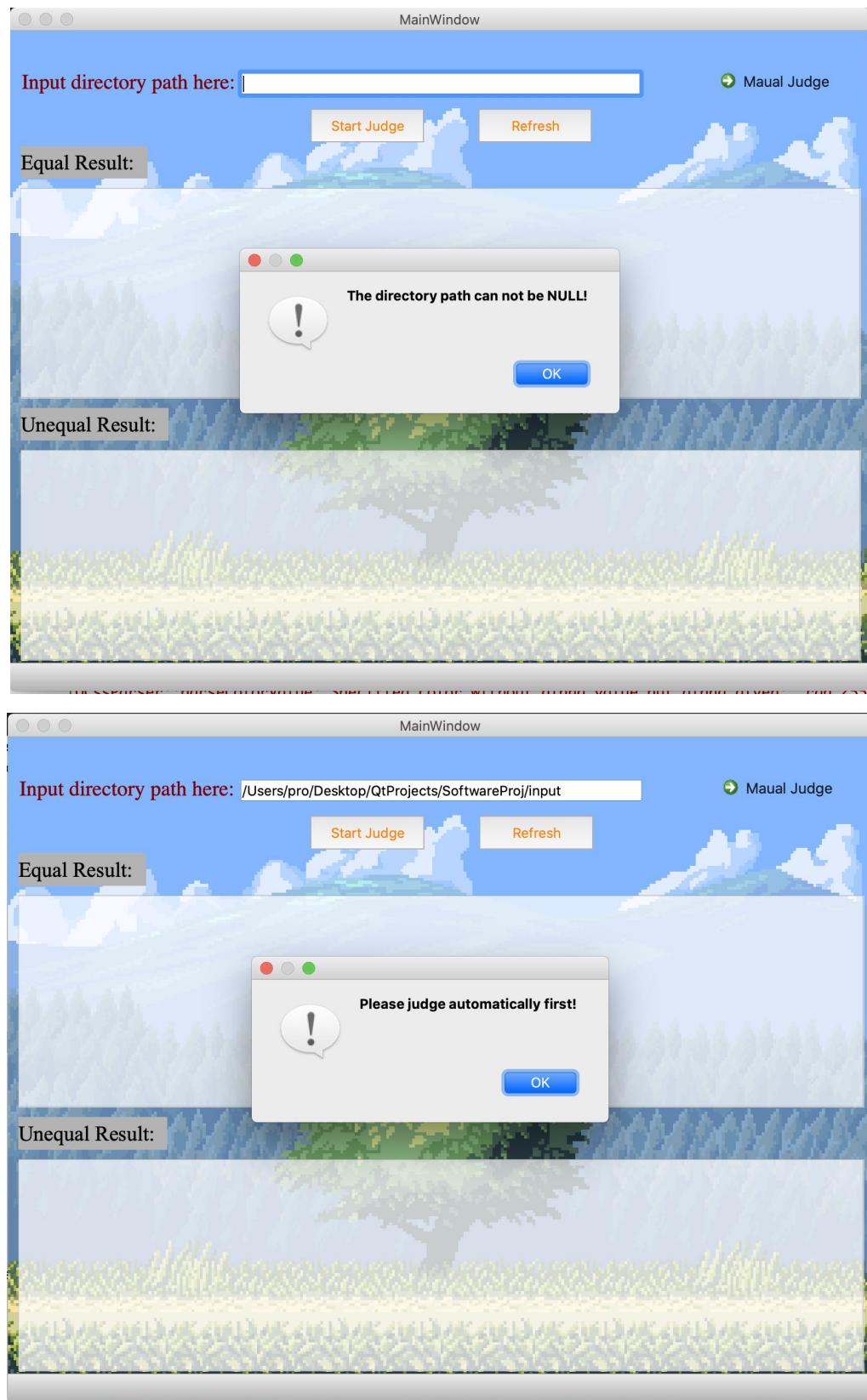
五、 拓展功能实现展示

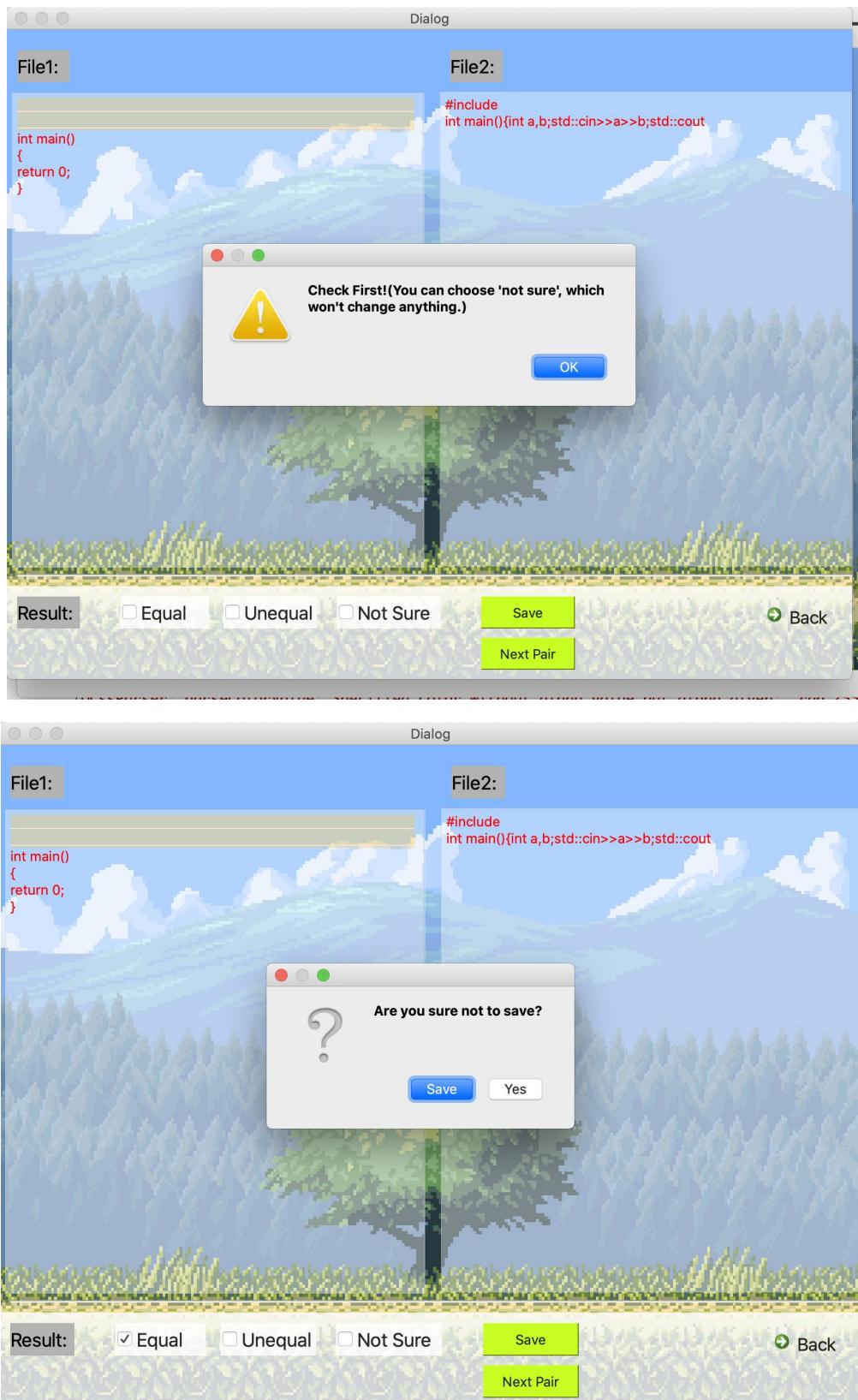
1. Diff 命令





2. 错误操作提示





- 未输入文件目录会有错误提示
- 必须先进行程序等价判断才可进入人工确认阶段
- 人工确认需要选择一项，否则会有错误提示
- 不保存会有提示是否确认不保存结果

3. 运行产生的 txt、exec、csv 文件可在本地查看

▼ SoftwareProj		今天 上午 9:38	--	文件夹
1.jpg		2022 年 9 月 17 日 上午 2:05	82 KB	JPEG 图像
bkg.jpg		2022 年 11 月 19 日 下午 1:27	20 KB	JPEG 图像
exec_out		今天 上午 9:38	--	文件夹
4A		今天 上午 9:38	--	文件夹
50A		今天 上午 9:38	--	文件夹
file.cpp		2022 年 10 月 25 日 下午 4:31	150 字节	C++ Source
file.h		2022 年 10 月 25 日 下午 4:31	137 字节	C Head...Source
handle.cpp		昨天 上午 11:19	14 KB	C++ Source
handle.h		今天 上午 9:10	755 字节	C Head...Source
input		今天 上午 9:38	--	文件夹
login.cpp		昨天 上午 8:56	348 字节	C++ Source
login.h		昨天 上午 8:56	302 字节	C Head...Source
login.ui		昨天 上午 9:52	593 字节	Qt UI File
main.cpp		昨天 下午 8:40	686 字节	C++ Source
mainwindow.cpp		昨天 下午 8:46	3 KB	C++ Source
mainwindow.h		昨天 下午 12:04	670 字节	C Head...Source
mainwindow.ui		昨天 上午 11:43	6 KB	Qt UI File
manual.cpp		昨天 下午 8:29	12 KB	C++ Source
manual.h		昨天 下午 7:28	931 字节	C Head...Source
manual.ui		昨天 下午 4:03	6 KB	Qt UI File
myResources.qrc		昨天 上午 11:47	181 字节	Qt Resource File
output		今天 上午 9:38	--	文件夹
equal.csv		今天 上午 9:38	5 KB	Comm...et (.csv)
unequal.csv		今天 上午 9:38	8 KB	Comm...et (.csv)

查看exec文件和输出结果txt文件

查看csv文件

实验评分: _____

指导教师签字: _____

年 月 日