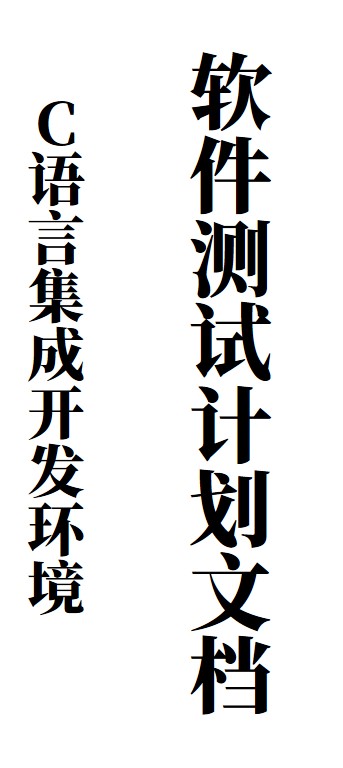
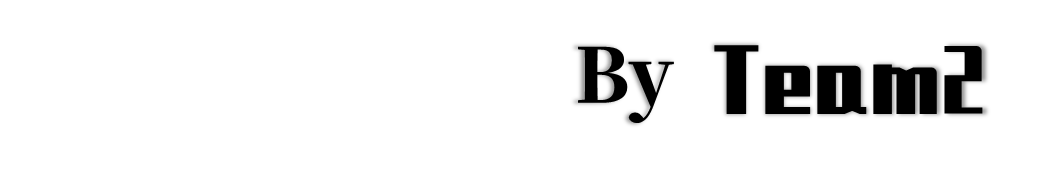
****

****



****

**目录**（鼠标左击可快速跳转到对应部分）

[1 范围 3](#_Toc118932938)

[1.1 标识 3](#_Toc118932939)

[1.2系统概述 3](#_Toc118932940)

[1.3 文档概述 4](#_Toc118932941)

[2 引用文档 4](#_Toc118932942)

[3 总体测评要求 5](#_Toc118932943)

[3.1 测试要求 5](#_Toc118932944)

[3.2 测试内容和测试方法 5](#_Toc118932945)

[3.3 数据采集要求 6](#_Toc118932946)

[3.4 被测软件评价准则和方法 7](#_Toc118932947)

[3.5 测试任务结束条件 7](#_Toc118932948)

[4 测试环境要求 7](#_Toc118932949)

[4.1测试环境 7](#_Toc118932950)

[4.2测试人员 8](#_Toc118932951)

[4.3开发方配合要求 8](#_Toc118932952)

[5 测试内容 9](#_Toc118932953)

[5.1 功能性测试 9](#_Toc118932954)

[5.2 性能效率测试 13](#_Toc118932955)

[5.3 兼容性测试 13](#_Toc118932956)

[5.4 易用性测试 14](#_Toc118932957)

[5.5 可靠性测试 14](#_Toc118932958)

[5.6 信息安全性测试 14](#_Toc118932959)

[5.7 维护性测试 15](#_Toc118932960)

[5.8 可移植性测试 15](#_Toc118932961)

[5.9 用户文档集测试 15](#_Toc118932962)

[6 项目管理 16](#_Toc118932963)

[6.1 组织与职责 16](#_Toc118932964)

[6.2 工作分工与进度安排 17](#_Toc118932965)

[6.3 跟踪与控制 17](#_Toc118932966)

1 范围

1.1 标识

1） 标题：基于QT的C语言集成开发软件测试计划；

2） 本文档适用的计算机软件：C语言集成开发软件；

3） 术语和缩略语：

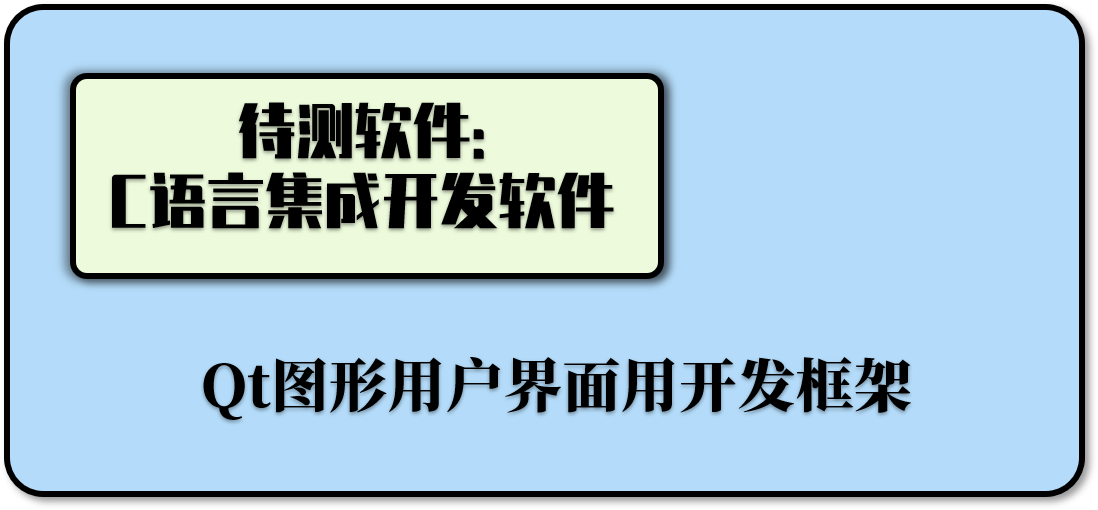
暂无

1.2系统概述

1.2.1 功能概述

C语言集成开发软件是一款简单的轻量级的C语言集成开发环境IDE，主要目的是为C语言编程者提供一个良好的编程环境，帮助C语言编程者更好编写代码。除了满足一款IDE基本的代码编辑、程序运行、编译与文件读写功能，还提供代码自动补全、自动缩进、全局查找与替换、关键字高亮和函数段折叠等进阶功能，可以极大地提升用户编写代码的效率和生产力。

软件外部交联图如下。



1.2.2 性能效率概述

C语言集成开发软件基于Qt平台开发，通过便捷调用轻量简洁的函数库可以进行只能语法分析，响应灵敏快速，在规定条件下，运行软件各项功能时，具有较短的处理时间，并且在运行过程中不会占用大量的CPU资源和内存，即使是在较低配置的机器上也能流畅地运行。除了能完成基本的对于普通单个程序文件的运行，还支持代码量较大的单个文件以及多个程序同时运行的情况。

1.2.3 兼容性概述

C语言集成开发软件在不同的软件系统中都有较好的相互集成的效果，在移植到不同系统平台的过程中不需要进行大规模地重新编写，软件开发以及部署的成本较低，经过一定的修改后可以稳定高效地运行在不同的系统平台，并且在多任务操作过程中，多个同时运行的软件不会出现冲突错误，可以方便地将数据共享给其他软件，不需要额外进行复杂地格式转换。

1.2.4 易用性概述

C语言集成开发软件有直观简洁的可视化操作界面，无需事先进行复杂地安装与配置过程，运行后即可直接编辑C语言程序。通过点击界面上的编译、运行等按钮图标即可一键实现相应功能。通过开发的进阶功能：代码自动补全、自动缩进、全局查找与替换、关键字高亮和函数段折叠等，大幅度地提升了用户在使用软件时的便捷度和代码编写的效率。

1.2.5 可靠性概述

C语言集成开发软件提供准确可靠的编译与调试功能，软件有较强的运行成熟性，可以避免掉运行过程中绝大多数的错误而导致的软件崩溃。并且在违背指定规则或者软件出现故障的情况下，仍然可以在运行时维持一定的基本响应功能。倘若已经受影响出现了软件失效的情况，也能在重新恢复后复原到崩溃前自动保存或记录的状态。

1.2.6 信息安全性概述

C语言集成开发软件具有基本的保护信息和数据的能力，以使未授权的人员或系统不能阅读或修改这些信息和数据，而不拒绝授权人员或系统对它们的访问。对用户、其他产品或者系统具有不同程度的授权类型与授权级别。软件会自行对不成功的操作记录进行预先定义，井明确规定达到该值时是否采取了具有规范性和安全性的措施来实现鉴别失败的处理。

1.2.7 维护性概述

C语言集成开发软件面对可能存在的修正、改进或者功能方向与规格变更都有较好的可维护性。软件的可理解性较高，开发过程中对特定功能的关键实现代码都做了清楚明了的注释说明，方便在维护过程中读懂开发时的思路。并且对不同的功能提供了模块化的编程，在维护过程中可以快速地定位到对应的类或代码段。

1.2.8 可移植性概述

C语言集成开发软件应用软件兼容性与系统兼容性较强，在移植到不同系统平台的过程中不需要进行大规模地重新编写，软件开发以及部署的成本较低。在目标环境下直接运行软件可执行代码的规范度也比较匹配，不需要为了适配不同的使用环境进行复杂的配置，同时也严格遵照了同类型软件相关的软件标准与约定，在同样的使用情况下可以方便地与同类软件相互替换。

1.2.9 软件其他信息

本次软件测试程序如下表。

**表1 被测软件程序清单**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **规模** | **开发环境/语言** | **运行平台** | **版本** | **重要度** |
| *3000行* | *Qt/C++* | *Windows* | *V3.0* | *重要* |

1.3 文档概述

本测试计划主要描述了测试要求、测试技术和方法、测试评价方法、测试数据采集要求、测试项终止要求、测试环境等资源要求、风险分析、进度安排等，为测试人员进行测试提供指导。

2 引用文档

1） GB/T 25000.51-2016《系统与软件工程 系统与软件质量要求和评价（SQuaRE） 第51部分：就绪可用软件产品（RUSP）的质量要求和测试细则》

2） BITSTC-JG27-03-V02软件测试计划编写规程（民品）

3 总体测评要求

3.1 测试要求

本次测试为系统测试，经过分析，测试内容包括9个方面，即功能性、性能效率、兼容性、易用性、可靠性、信息安全性、维护性、可移植性及用户文档集。具体见表。

针对本次测试的9个测试内容，采用的主要测试用例设计方法有：功能分解法、等价类划分法、边界值分析法、正交实验设计法、猜错法等。不同用例中所使用的方法将在每个用例中具体说明。

对测试过程中发现的软件缺陷进行一次回归测试，测试的最终结果以回归后的测试结果为准。

**表2 测试内容说明**

|  |  |  |
| --- | --- | --- |
| **测试内容名称** | **测试内容标识** | **测试内容描述** |
| 功能性测试 | GN | 产品的功能正确性、功能实现的完整性、功能的计算准确性、安全性等。 |
| 性能效率测试 | XN | 测试任务响应时间和资源利用率等。 |
| 兼容性测试 | JR | 测试软件间的共存性和互操作性。 |
| 易用性测试 | YY | 测试软件产品的界面交互能力、帮助文档的易理解性、易学性和易操作性等。 |
| 可靠性测试 | KK | 包括容错性、数据保护、系统在测试期间运行稳定性等。 |
| 信息安全性测试 | XA | 测试软件运行期间数据的保密性、完整性和真实性等。 |
| 维护性测试 | WH | 软件模块化、易分析性、易修改性、易测试等。 |
| 可移植性测试 | YZ | 测试软件的易安装性、易替换性等。 |
| 用户文档集测试 | WD | 测试用户文档的完备性、正确性、一致性、可操作性、易学易理解性等。 |

3.2 测试内容和测试方法

**表3 测试方法说明**

|  |  |
| --- | --- |
| **测试内容名称** | **测试方法** |
| 功能性测试 | 根据需求规格说明，采用等价类划分、边界值分析、错误推测、场景法等黑盒测试技术，通过设计覆盖全部软件产品功能实现的测试用例的方法，从功能实现的功能正确性、功能实现的完整性、功能的计算准确性、安全性等方面对用户需求的软件产品全部功能性进行质量测试，并将功能性检测结果与用户需求中的功能要求比较，评价该软件产品功能是否符合用户需求和文档要求。 |
| 性能效率测试 | 根据需求规格说明，采用压力测试、负载测试、容量测试、强度测试等黑盒测试技术，通过使用通用或专用测试工具及设备和设计测试用例的方法，从响应时间、事物成功率、资源利用率等方面对软件产品的效率进行质量测试，并将效率测试结果与用户需求中的效率要求比较，评价软件产品效率是否符合有那刚好需求和文档要求。 |
| 兼容性测试 | 根据被测软件产品的特点，采用跨平台测试、数据设计测试等黑盒测试技术，通过设计测试用例，从硬件兼容性、软件兼容性、数据兼容性、网络协议兼容性等方面对软件产品的兼容性质量特性进行测试，并将兼容性的测试结果与用户需求中的兼容性要求比较，评价软件产品兼容性是否符合用户需求和文档要求。 |
| 易用性测试 | 根据需求规格说明，采用验证软件执行的各种操作/输入/问题/消 息/结果的易理解性、检查文档和帮助信息、模拟演示界面的互操作性等黑盒测试技术，通过验证、检查和实际测试用例的方法，从易理解性、易学性、易操作性等方面对软件产品的易用性进行质量测试，并将易用性测试结果与用户需求中的易用性要求比较，评价软件产品的易用性是否符合用需求和文档要求。 |
| 可靠性测试 | 根据需求规格说明采用错误推测方法、边界分析、错误恢复、恢复性测试和稳定性测试等黑盒测试技术，通过设计软件产品异常处理及风险处理测试用例的方法，从容错性、数据保护、运行稳定性等方面对软件产品的可靠性进行质量测试，并将可靠性测试结果与用户需求中的可靠性要求比较，评价软件产品的可靠性是否符合用户需求。 |
| 信息安全性测试 | 根据被测软件产品的特点，选取正常或非正常测试用例，测试软件产品授权访问、权限控制等实际情况，并将新鲜安全性测试结果与软件产品新鲜安全性要求进行比较，评价信息安全的符合性。 |
| 维护性测试 | 根据被测软件的特点，采用接口测试技术、流程控制测试、数据流测试和模块黑盒测试技术，同验证、检查和测试用例的方法，从易分析性、易改变性、稳定性、易测试性和维护的依从性等方面对软件产品的维护性进行质量测试，滨江维护性测试结果与软件维护性测试要求进行比较，评价软件维护性的符合性。 |
| 可移植性测试 | 根据被测软件的特点，选取产品说明中列出的所有支持平台和系 统，对软件进行安装，运行、卸载，并将可移植性测试结果与软件的可移植性要求进行比较，评价软件可移植性的符合性。 |
| 用户文档集测试 | 对被测软件的用户文档内容进行测试，重点验证检查其完备性、正确性、一致性、易理解性、易学性、可操作性。 |

3.3 数据采集要求

测试过程中要采集如下方面的数据，以评价被测软件和测试质量：

1） 被测软件数据：代码规模、测试项个数、可测项个数；

2） 用例数据：生成的测试用例总数，每个测试项中不同测试类型的用例数，执行了的用例数，未执行的用例数；

3） 缺陷数据：测试发现的缺陷数，并按类型和重要度进行划分；典型缺陷及影响；

4） 管理数据：工作量、进度；

3.4 被测软件评价准则和方法

在软件测试报告中从如下方面对软件进行定性评价：

1） 功能性方面：系统中存在未修改的“关键缺陷”大于等于1个或存在未修改的“重要缺陷”大于等于3个时视为“不通过”，其它情况视为“通过”；

2） 性能效率方面：性能效率符合指标要求则“通过”，否则“不通过”；

3） 兼容性方面：同功能性方面；

4） 易用性方面：同功能性方面；

5） 可靠性方面：同功能性方面；

6） 信息安全性方面：同功能性方面；

7） 维护性方面：同功能性方面；

8） 可移植性方面：同功能性方面；

9） 用户文档集方面：同功能性方面；

10） 系统总体评价：系统各项质量要求均为“通过”时则系统视为“通过”，除功能性要求外只有1项质量要求为“不通过”，其它为“通过”时则系统也视为“通过”；功能性要求为“不通过”是则系统视为“不通过”，除功能性要求外多于1项质量要求为“不通过”则系统视为“不通过”。

3.5 测试任务结束条件

完成了如下工作则测试任务结束：

1） 所有测试项都达到了测试终止要求，且发现的每个软件缺陷都作了归零处理（进行了修改并通过回归测试，或没修改但给出了降低风险的措施，或进行风险分析后可以接受不修改带来的风险故不作修改。）且提交了相应的测试文档；

2） 完成了测试协议中所规定的所有工作；

3） 通过了测试相关评审，且按照评审意见中的建议对工作进行了完善。

4 测试环境要求

4.1测试环境

4.1.1 软件项

**表4 软件项**

|  |  |  |  |
| --- | --- | --- | --- |
| **序号** | **软件项名称** | **版本** | **用途** |
| 1 | C语言集成开发软件 | 代码审查后版本 | 被测软件 |
| 2 | Qt图形用户界面应用开发框架 | V6.1.2 | 应用软件编译器、代码连接器 |

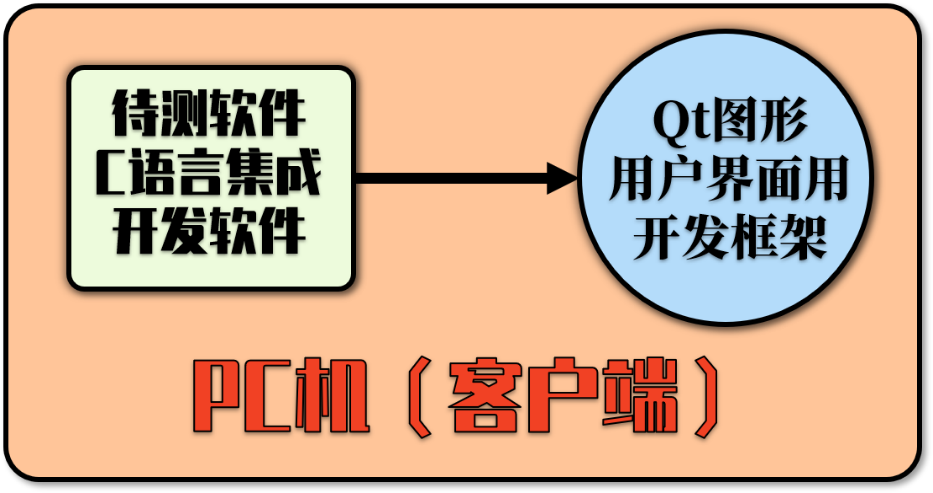
4.1.2 硬件和固件项

**表5 硬件和固件项**

|  |  |  |  |
| --- | --- | --- | --- |
| **序号** | **硬件和固件项名称** | **用途** | **最低配置要求** |
| 1 | PC机（客户端） | 作为载体承担待测软件的运行与测试 | cpu：i5-5200U@2.20GHz  硬盘：1TB  内存：8G  分辨率：1366x768  显卡：930M |

4.1.3 测评运行环境

根据测试要求，建立C语言集成开发软件配置项测试环境，测试环境示意图如下：



4.2测试人员

杨宜松 张 桢

刘睿泽 吕泽超

陈晓璐 梁 坤

4.3开发方配合要求

经协商，开发方在如下阶段进行相关配合：

**表6 开发方配合内容**

|  |  |  |
| --- | --- | --- |
| **阶段** | **内容** | **开发方人员** |
| 测试需求分析 | 介绍软件功能 | 陈晓璐 |
| 测试策划 | 各时间节点和配合要求讨论 | 刘睿泽 |
| 测试设计和实现 | 提供检测仪器，用例讨论 | 吕泽超、刘睿泽 |
| 测试执行 | 结果确认，确认问题签字，修改软件，填写更改单，回归测试配合 | 梁坤 |
| 测试总结 | 确认评审时间 | 刘睿泽 |

5 测试内容

5.1 功能性测试

5.1.1 菜单检查

|  |  |  |  |
| --- | --- | --- | --- |
| 测试项名称 | 基本菜单界面 | 测试项标识 |  |
| 追踪关系 |  | | |
| 测试项描述 | 菜单栏，是由下拉菜单组成的，点击每一个菜单都会出现一个下拉菜单，里面包含具体的功能。菜单栏最重要的是文件、编辑两个菜单。  工具栏，是由图标组成的，每一个图标代表一个常用的功能，点击图标就可触发响应功能。工具栏最重要的是新建文件，打开文件，保存文件与编译运行。 | | |
| 软件设计约束 | 无。 | | |
| 测试内容 | 1.直接运行程序，点击菜单栏的具体菜单，查看是否有下拉菜单；  2.点击功能，看能否实现该功能。  3.将鼠标悬停于图标，查看文字；鼠标点击图标查看是否能进行功能跳转。  4.首先查看左部的文件视图栏和底部的编译输出框是否可以正常显示；其次点击按钮查看是否能隐藏和再现  5.打开任意包含中英文和符号的文件，尝试修改，查看显示是否正确；查看行号对应是否正确 | | |

5.1.2 文件基本操作检查

|  |  |  |  |
| --- | --- | --- | --- |
| 测试项名称 | 文件操作 | 测试项标识 |  |
| 追踪关系 |  | | |
| 测试项描述 | 包含在菜单操作中，有 保存/打开/另存/关闭 功能。保存需要指定保存的路径，文件名，文件类型；打开需要选择打开哪个文件；另存需要指定保存的路径，文件名，文件类型；关闭需要提示用户是否需要保存。  在用户创建一个类项目时，可以自动生成类的头文件和类的函数的.cpp文件，可以进行命名，之后再在源文件中创建一个包含主函数的文件即可。  支持用户在头文件中创建.h文件，在源文件中创建.cpp文件。 | | |
| 软件设计约束 | 无。 | | |
| 测试内容 | 1.打开或新建多个文件；  2.打开文件关闭；打开文件修改后关闭  3.点击文件菜单下的新建按钮  4.目录中右键点击文件夹新建文件，点击新建文件，输入文件名，选择文件类型，点击确定，在该路径下新建文件。  5.键点击目录中的任意文件夹弹出自定义的右键菜单，点击新建文件夹，弹出新建文件夹弹窗，输入文件夹名字，点击确定，文件夹被新建在当前目录下，并添加到本地。  6.点击建立一个文件夹。  7.点击menubar中的文件，弹出下拉菜单，点击打开，弹出文件选择页面，选择文件，点击打开，文件所在路径下的文件展示在侧边目录栏中，并根据文件类型添加图标，文件内容显示在右侧编辑框中。  8.点击menubar中的关闭文件。  9.右键点击工作路径文件夹关闭项目，点击关闭项目，工作路径消失。  10.点击menubar中的文件，点击另存，弹出提示框提示你输入文件名和文件类型，输入文件名和保存类型点击保存，弹出提示框显示保存成功。  11.编辑文件内容后点击保存，检查本地磁盘中是否已保存文件。  12.右键点击任意一个文件，点击删除文件，删除该文件。  13.鼠标光标指向一个文件夹，点击右键，弹出定制化弹窗，点击删除文件夹，文件夹在目录中消失，并在本地删除。  14.右键点击文件，点击打开所在文件夹，弹出文件资源管理器，显示该路径下的文件。  15.双击目录中的文件，显示在右侧编辑框中。  16.点击右上角关闭按钮。 | | |

5.1.3 文本基本操作检查

|  |  |  |  |
| --- | --- | --- | --- |
| 测试项名称 | 文本操作 | 测试项标识 |  |
| 追踪关系 |  | | |
| 测试项描述 | 选中文本段，按下“Ctrl+C”进行复制，按下“Ctrl+V”进行粘贴，按下“Ctrl+X”进行剪切，并在“编辑”菜单中有实体的“复制”、“粘贴”、“剪切”按钮。  对任意字符串可以进行插入、修改和删除操作，将光标移到需要修改的地方点击，即可进行增、删、改、查等操作，按下“Ctrl+Z”可以撤销上一步操作。  提供全局查找功能，对于选中的字符或字符串，点击“全局查找”可以找出全文中含有多少个该字符串，并高亮所有字符串。  提供顺序查找功能，对于选中的字符或字符串，可首先高亮显示全文第一个该字符串，点击“next”按钮可找到并高亮下一个字符串。  提供全局替换功能，选中某字符或字符串后，右键并选择“全部替换”，在任意一处字符串处修改；或者点击“编辑”菜单中的“替换”，在输入框修改，即可将全文的原字符串替换为新字符串，并且都标亮。 | | |
| 软件设计约束 | 无。 | | |
| 测试内容 | 1.点击图标或者menubar中的粘贴。  2.选中部分文字，点击图标剪切或者menubar中展开的剪切  3.编写一段文本，点击全局查找。  4.编写一段文本，点击查找上一个，观察具体查找的过程。主要观察查找的高亮。  5.编写一段文本，点击查找下一个，观察具体查找的过程。主要观察查找的高亮。  6.编写一段文本，选定所要替换的字符与替换之后的字符点击全局替换，观察替换是否成功。  7.编写一段文本，点击查找，如果查找到了，字符会显示黄色的高亮，之后写好你想要的字符，点击替换当前字符。 | | |

5.1.4 编译运行检查

|  |  |  |  |
| --- | --- | --- | --- |
| 测试项名称 | 编译操作 | 测试项标识 |  |
| 追踪关系 |  | | |
| 测试项描述 | 在选择好特定的编译器后点击“编译”按钮（也可按下快捷键F5），会调用对应的编译器将用户组织编写的文本形式的源代码转化为可执行文件（.exe）。  在这个过程中，会同时在编译日志项目栏中输出编译日志，日志中包括调用的编译器、警告数、错误数、编译时间。  鼠标点击行号左侧可以添加断点。  鼠标点击Debug按钮启动Debug程序。  点击程序中的变量可以在控制台中显示Debug过程中变量的值。 | | |
| 软件设计约束 | 无。 | | |
| 测试内容 | 1.编写一段正确的代码，点击运行。  2.编写一段错误的代码，点击运行。  3.编写一段正确的代码，或使用已经通过编译的文件，点击debug。  4.编写一段正确的代码，或使用已经通过编译的文件，点击增加，删除，查看断点按钮。  5.编写一段正确的代码，或使用已经通过编译的文件，增加断点后运行程序。  6.编写一段正确的代码，或使用已经通过编译的文件，进行调试并查看变量。 | | |

5.1.5 代码编写检查

|  |  |  |  |
| --- | --- | --- | --- |
| 测试项名称 | 代码编写 | 测试项标识 |  |
| 追踪关系 |  | | |
| 测试项描述 | 识别相应关键字，如：int。  能自动填充关键字，如输入“in”，光标下方会有提示有哪些，按下“Tab”会自动补足。  对不符合格式的关键字，如“include”少写了一个n，下方用红色波浪线标注，光标移至错误处时会有错误提示，如“iclude”:unknown word  在函数体、循环体、判断语句内部写代码的时候，所在的外层“{}”会有光标停留显示，提示编程人员其所处位置。  用户打出一个“{”，程序自动匹配“}”。  检查括号的逻辑正确性：如果用户打出“{”但是没有“}”与之匹配，需要标红以提醒用户。  在“{}”中再次按下“}”键支持光标向右移动一位，退出括号。  当光标放在括号位置时，该括号以及与该括号匹配的括号会有光标提示。  显示行号。  同一个函数，”while”, “for”循环体，以及判断语句的开头部分代码左侧会有折叠按钮  点击后，函数以及循环体部分会折叠到函数或循环体第一行后，以”…”形式呈现 | | |
| 软件设计约束 | 无。 | | |
| 测试内容 | 1.点击未被折叠的左大括号所在行行号，再点击被折叠的左大括号所在行行号，测试多组不同位置不同嵌套等级的大括号。  2.输入三个字符，观察是否有提示出现，选择一个提示按下回车，观察是否可以自动补全。  3.输入“{”，观察其是否自动补全；在输入回车，观察其是否自动缩进。  4.分别测试输入“[”、“{”、“(”，观察在输入框内是否补全了后括号，且光标是否放置于括号中间。  5.将光标位置至于括号处，观察该括号以及与其匹配的括号是否出现光标提示。另外，将光标移到其他位置或者字符串发生变化时，该提示是否消失。  6.编写代码后，增加几行，减少几行代码，查看行号的变化是否正确。  7.点击代码折叠函数，或while/for循环体，检查折叠效果。 | | |

5.1.6 关键字高亮检查

|  |  |  |  |
| --- | --- | --- | --- |
| 测试项名称 | 关键字高亮 | 测试项标识 |  |
| 追踪关系 |  | | |
| 测试项描述 | 识别相应关键字，如：int，根据识别到的关键字，并根据配色方案进行相应的上色。 | | |
| 软件设计约束 | 无。 | | |
| 测试内容 | 1.输入对应的字符，观察是否有高亮显示，是否正确。  2.输入特殊符号字符，观察是否有高亮显，是否正确。  3.输入头文件字符，观察是否有高亮显示，是否正确。  4.输入数字字符，观察是否有高亮显示，是否正确。  5.输入单行和多行注释，观察是否有高亮显示，是否正确。  6.输入引号及字符串，观察是否有高亮显示，是否正确。  7.输入函数，观察是否有高亮显示，是否正确。  8.输入非关键字，观察是否有高亮显示。 | | |

5.1.7 鼠标键盘操作检查

|  |  |  |  |
| --- | --- | --- | --- |
| 测试项名称 | 鼠标键盘操作 | 测试项标识 |  |
| 追踪关系 |  | | |
| 测试项描述 | 可以右击鼠标打开菜单  滚轮与ctrl配合可以缩小放大屏幕 | | |
| 软件设计约束 | 无。 | | |
| 测试内容 | 1.进入程序，鼠标右键单击菜单。  2.先是不按住Ctrl单独滑动鼠标滚轮上下，然后按住Ctrl后单独滑动鼠标滚轮上下，测试多组不同位置不同幅度的鼠标滚轮滑动。检查放大和缩小。 | | |

5.1.8 切换主题操作检查

|  |  |  |  |
| --- | --- | --- | --- |
| 测试项名称 | 主题切换 | 测试项标识 |  |
| 追踪关系 |  | | |
| 测试项描述 | 根据需求进行主题切换，包括颜色和图标的切换。 | | |
| 软件设计约束 | 无。 | | |
| 测试内容 | 1.在菜单栏中点击“更换主题”，选择不同风格，观察主题颜色与图标的改变。 | | |

5.2 性能效率测试

|  |  |  |  |
| --- | --- | --- | --- |
| 测试项名称 | 性能效率测试 | 测试项标识 | XN |
| 追踪关系 |  | | |
| 测试项描述 | 支持单文件编辑，文本长度不超过2000行。  装入/保存2000行源程序时间小于3秒。  基本操作（插入/删除/翻页/查找等）响应无明显延迟。  全文自动替换响应时间小于3秒。 | | |
| 软件设计约束 | 无。 | | |
| 测试内容 | 针对每一种性能性能需求分别进行测试，对时间进行检查。 | | |

5.3 兼容性测试

|  |  |  |  |
| --- | --- | --- | --- |
| 测试项名称 | 兼容性测试 | 测试项标识 | JR |
| 追踪关系 |  | | |
| 测试项描述 | 可以在windows，mac，linux系统上运行  支持运行用户的.c文件 | | |
| 软件设计约束 | 无。 | | |
| 测试内容 | 1. 在windows，mac，linux系统上分别运行软件，检查功能是否可以正常执行 2. 运行用户的.c文件，查看是否有运行结果，结果是否正确 | | |

5.4 易用性测试

|  |  |  |  |
| --- | --- | --- | --- |
| 测试项名称 | 易用性测试 | 测试项标识 | YY |
| 追踪关系 |  | | |
| 测试项描述 | 直观，人机交互效果好的界面。  操作便捷，功能完善。 | | |
| 软件设计约束 | 无。 | | |
| 测试内容 | 1.用户界面的友好性：界面的简洁性如何，是否为系统管理员提供简洁、直观、友好的图形化管理界面，方便操作。  2.易操作性：操作的难易程度，对主要或常用功能应该提供快捷方式。 | | |

5.5 可靠性测试

|  |  |  |  |
| --- | --- | --- | --- |
| 测试项名称 | 可靠性测试 | 测试项标识 | KK |
| 追踪关系 |  | | |
| 测试项描述 | 用户的文件数据要求必须完整读入编译  软件在运行期间不会崩溃  对于软件异常进行基本处理 | | |
| 软件设计约束 | 无。 | | |
| 测试内容 | 1.数据完整性：数据完整性审查，编译用户的文件检查编译结果是否为完整文件。  2.测试不同状态下软件是否会崩溃。  3.抛出异常，检查软件是否可以自动处理异常 | | |

5.6 信息安全性测试

|  |  |  |  |
| --- | --- | --- | --- |
| 测试项名称 | 信息安全性测试 | 测试项标识 | XA |
| 追踪关系 |  | | |
| 测试项描述 |  | | |
| 软件设计约束 | 无。 | | |
| 测试内容 | 信息安全性陈述：检查产品说明是否包括有关信息安全性的陈述， 包括保密性、完整性、抗抵赖性、可核查性、真实性以及信息安全 性的依从性，并经书面形式展示可验证的依从性证据。 | | |

5.7 维护性测试

|  |  |  |  |
| --- | --- | --- | --- |
| 测试项名称 | 维护性测试 | 测试项标识 | WH |
| 追踪关系 |  | | |
| 测试项描述 | 软件模块和功能具有可重用性、易分析性、易改变性、易测试性，并且在其他软件中复用性强。 | | |
| 软件设计约束 | 无。 | | |
| 测试内容 | 维护性陈述：检查产品说明是否包含有关维护的陈述，包括模块 化、可重用性、易分析性、易改变性、易测试性以及维护性的依从 性，并经书面形式展示可验证的依从性证据。  维护信息：检查产品说明是否有用户所需的维护信息。  用户修改：当软件能由用户作适应性修改时，检查产品说明是否标 识用于修改的工具或规程及其使用条件。 | | |

5.8 可移植性测试

|  |  |  |  |
| --- | --- | --- | --- |
| 测试项名称 | 可移植性测试 | 测试项标识 | YZ |
| 追踪关系 |  | | |
| 测试项描述 | 软件可以安装和移植。 | | |
| 软件设计约束 | 无。 | | |
| 测试内容 | 可移植性陈述：检查软件产品说明是否包含有关可移植性的陈述， 包括适应性、易安装性、已替换性以及可移植的依从性，并经书面 形式展示可验证的依从性证据。  安装规程：检查产品说明中是否提供安装规程信息。 | | |

5.9 用户文档集测试

|  |  |  |  |
| --- | --- | --- | --- |
| 测试项名称 | 用户文档集测试 | 测试项标识 | WD |
| 追踪关系 |  | | |
| 测试项描述 | 提供用户文档，软件使用说明书等相关文档 | | |
| 软件设计约束 | 无。 | | |
| 测试内容 | 规范性：用户文档描述规范，有版本控制修改记录。  符合性：用户文档与需求和设计文档的符合程度。  完整性：用户手册内容基本完整，对具体操作的说明比较详细。  一致性：用户手册的描述与软件的实际功能基本一致，对重要功能 的说明比较全面，用户手册中具有产品版本号描述。  易理解程度：用户手册对操作有图例和文字说明，较易理解。 | | |

6 项目管理

6.1 组织与职责

项目经理要对测试组成员划分角色，并明确其职责，除项目经理外，测试组中的角色还应包 括测试人员、测试监督人员、质量保证人员、测试组配置管理人员、测试组资料管理员、测试组设备管理员、测试组被测件管理员。各角色职责如下表，各角色可允许兼职。

表7 测试成员分工及职责

|  |  |  |
| --- | --- | --- |
| 角色 | 人员姓名 | 职责 |
| 分管项目领导 |  | * 对软件测试项目的全过程进行策划和管理，负责编写软件测评项目工作计划和项目总结报告； * 协同配置管理员和质量保证人员制定《配置管理计划》和《质量保证计划》； * 负责根据测试结果进行软件评价； * 定期听取各测试组软件质量保证 (SQA) 人员和软件配置管理 (SCM)人员对测试过程质量问题的报告，并采取相应的纠正措施，以使测试过程受控； * 控制本项目的进度、相关费用并决定对项目组成员的奖惩； * 组织系统测试相关评审，组织基线变更的评审； * 对项目的保密负责。 |
| 项目经理 |  | * 对承担的测试任务进行策划，并负责制定软件测试计划和测试说明； * 负责组织实施软件测评工作，在实施过程中根据测评实施情况和项目组SQA和SCM人员的意见及时对测评过程中出现的偏离提出纠正措施，确保测评过程受控； * 组织并参加测评报告的编制，负责对被测软件进行评价和说明； * 及时向测评室主任报告测评项目进展状况及存在问题； * 与开发方沟通并落实评审； * 落实各项保密措施； * 组织测试资料的归档 (含电子文档)， 被测件的退还工作。 |
| 测试人员 |  | * 进行软件测评需求分析，完成测试设计、生成测试用例，搭建测试环 境； * 测试的实施，并在过程中认真贯彻质量手册、过程文件和作业指导书 的有关规定； * 认真记录测试结果，填写质量记录，收集典型缺陷，及其他规定信息，编制问题报告； * 进行评审工作的准备； * 各项保密措施的实施。 |
| 测试监督员 |  | 按照测试计划和说明情况跟踪测试工作实施情况，主要包括：   * 进度、工作量、测试发现问题情况、测试充分性、遗留问题； * 跟踪和评价测试工作的有效性； * 跟踪和评价测试工具的使用情况； * 对测试数据、质量记录的真实性和公正性进行监督、检查； * 对测评过程中出现的质量问题及时向测试组长报告。 |
| 配置管理员 |  | * 与项目主管协商制定测试组软件配置管理计划； * 测试过程中对开发库进行管理。 |
| 质量保证人员 |  | * 与项目主管协商制定测试组质量保证计划； * 负责对测评项目的整个实施过程进行质量监督，按照测试各阶段SQA检 查单检查过程的符合性； * 对测试数据、质量记录的真实性和公正性进行监督、检查； * 对测评过程中出现的质量问题及时向软件质量保证室主任报告。 |
| 被测件管理员 |  | * 测试过程中被测件的发放、管理、收回工作及被测件保密工作的落实。 |
| 设备管理员 |  | * 负责测试环境的标识/状态标识管理工作； * 负责测试环境的检验、校准和例外处理工作。 |
| 用户代表 |  | * 协调、监督测试工作，审核不修改的软件问题。 |

6.2 工作分工与进度安排

进行工作分解，并制定进度安排，见表8。

**表8 工作进度安排**

|  |  |  |  |
| --- | --- | --- | --- |
| 工作任务 | 负责人 | 工作量 | 完成日期 |
| 确定测试目标项目 | 刘睿泽 |  | 2022/11/7 |
| 初步完成测试计划文档 | 杨宜松，吕泽超，梁坤 |  | 2022/11/10 |
| 测试执行 |  |  |  |
| 测试总结 |  |  |  |

6.3 跟踪与控制

在测试过程中测试监督人员对测试按如下方式进行监督：

**表9 测试监督内容与方式**

|  |  |  |  |
| --- | --- | --- | --- |
| 阶段 | 跟踪内容 | 跟踪方式 | 跟踪频度 |
| 测试需求及测试策划 | 测试项划分情况 | 例会 | 一周两次 |
| 测试设计、文档审查及代码审查 | 用例设计的情况和进度；文档审查、代码审查的情况和进度；环境构建情况和进度；存在的问题；测试人员 | 例会 | 一周两次 |
| 测试执行 | 每日已测用例数；每日发现问题数统计；用例调整、环境问题。 | 总结会 | 两周一次 |
| 测试总结 | 信息的完整性。 | 例会 | 一周两次 |

过程跟踪表参照《软件测评过程监督程序》中的跟踪表，其中的具体工作内容可根据具体情况细化。

在生成用例中，监督人员负责审核相应的用例的合理性和充分性，单元测试具体分工参见表10。

**表10 测试分工与需求追踪的关系**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 需求追踪  (软件详细设计) | 测试项名称  (单元名称) | 测试项标识 | 用例生成人员 | 审核人员 |
| 5.2.1.1 | 弹出菜单 |  |  |  |
| 5.2.1.2 | 快捷缩放 |  |  |  |
| 5.2.1.3 | 函数折叠 |  |  |  |
| 5.2.1.4 | 全局查找 |  |  |  |
| 5.2.1.5 | 查找上一个/下一个 |  |  |  |
| 5.2.1.6 | 文本自动补全 |  |  |  |
| 5.2.1.7 | 括号缩进 |  |  |  |
| 5.2.1.8 | 更换主题 |  |  |  |
| 5.2.1.9 | 编译运行 |  |  |  |
| 5.2.1.10 | Debug |  |  |  |
| 5.2.1.11 | 括号补全 |  |  |  |
| 5.2.1.12 | 括号匹配 |  |  |  |
| 5.2.1.14 | 关键字高亮 |  |  |  |
| 5.2.1.15 | 文件操作 |  |  |  |
| 5.2.1.16 | 多文件操作 |  |  |  |
| 5.3 | 性能需求 |  |  |  |