**MsEditor需求文档**

目录

[（一） 需求理解 2](#_Toc53644289)

[（二） 界面设计 2](#_Toc53644290)

[一． 初始界面 2](#_Toc53644291)

[二． 文本编辑界面 3](#_Toc53644292)

[三． 编译运行界面 4](#_Toc53644293)

[四． 搜索替换界面 4](#_Toc53644294)

[五． Debug界面 5](#_Toc53644295)

[（三） 功能及操作设计 6](#_Toc53644296)

[一． 多文件操作 6](#_Toc53644297)

[二． 编译运行 6](#_Toc53644298)

[三． 项目操作 6](#_Toc53644299)

[四． 编码 7](#_Toc53644300)

[五． 关键字高亮 7](#_Toc53644301)

[六． 函数操作 7](#_Toc53644302)

[七． 搜索替换 7](#_Toc53644303)

[八． 行号显示 8](#_Toc53644304)

[十． 关键字自动补全 9](#_Toc53644305)

[十一． 隐藏注释 9](#_Toc53644306)

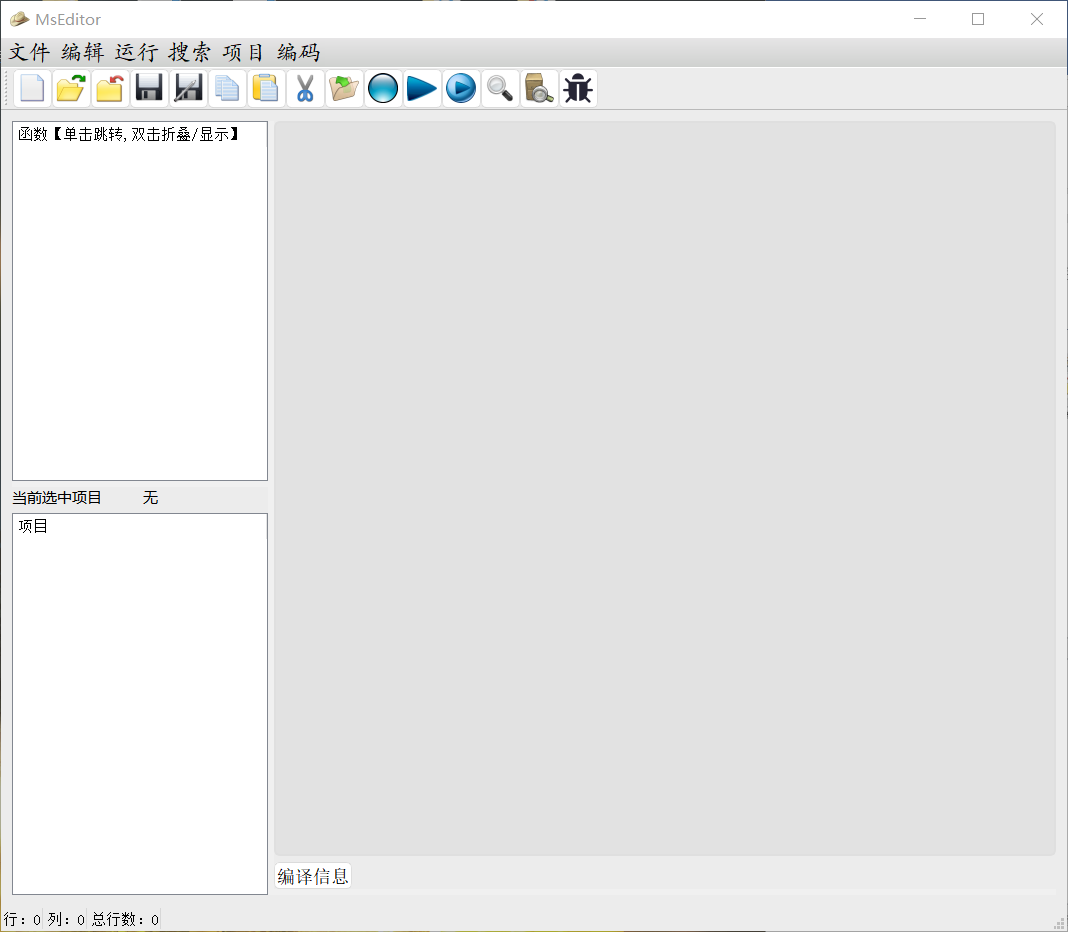
[十二． Debug功能 9](#_Toc53644307)

# 需求理解

我们小组对于需求的理解是，开发出一款能够满足几乎所有基础功能需求、并且便捷易用的集成开发环境软件。我们决定参考Dev-C++的UI及功能设计，同时使产品具有自己的特色。

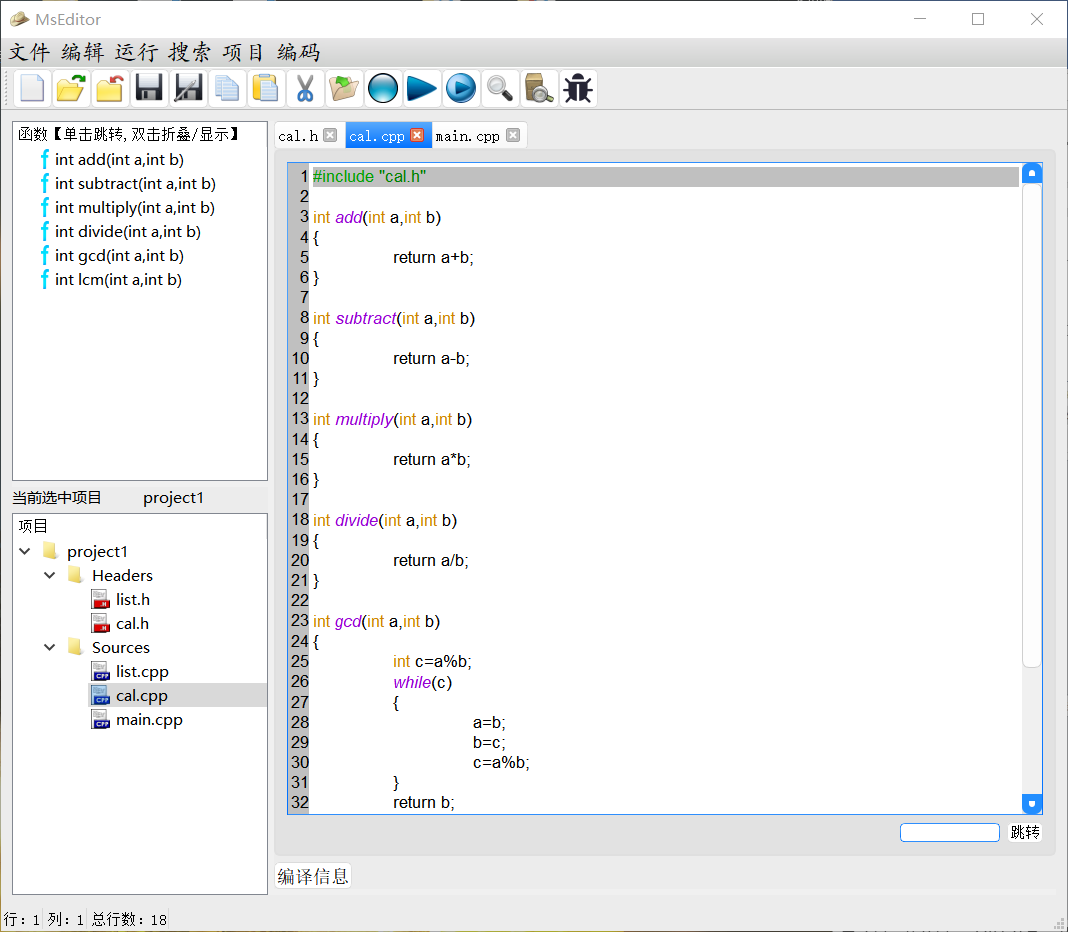
# 界面设计

## 初始界面



打开软件之后，首先出现的是初始界面。上面是菜单栏和工具条。此时还没有打开任何文件，因此右侧的文本编辑框是一片空白。编译信息框藏在文本编辑框下方，在编译运行时会自动弹出，也可以手动将其弹出。左侧上面是函数栏，下面是项目栏。最下方的是状态栏。

## 文本编辑界面



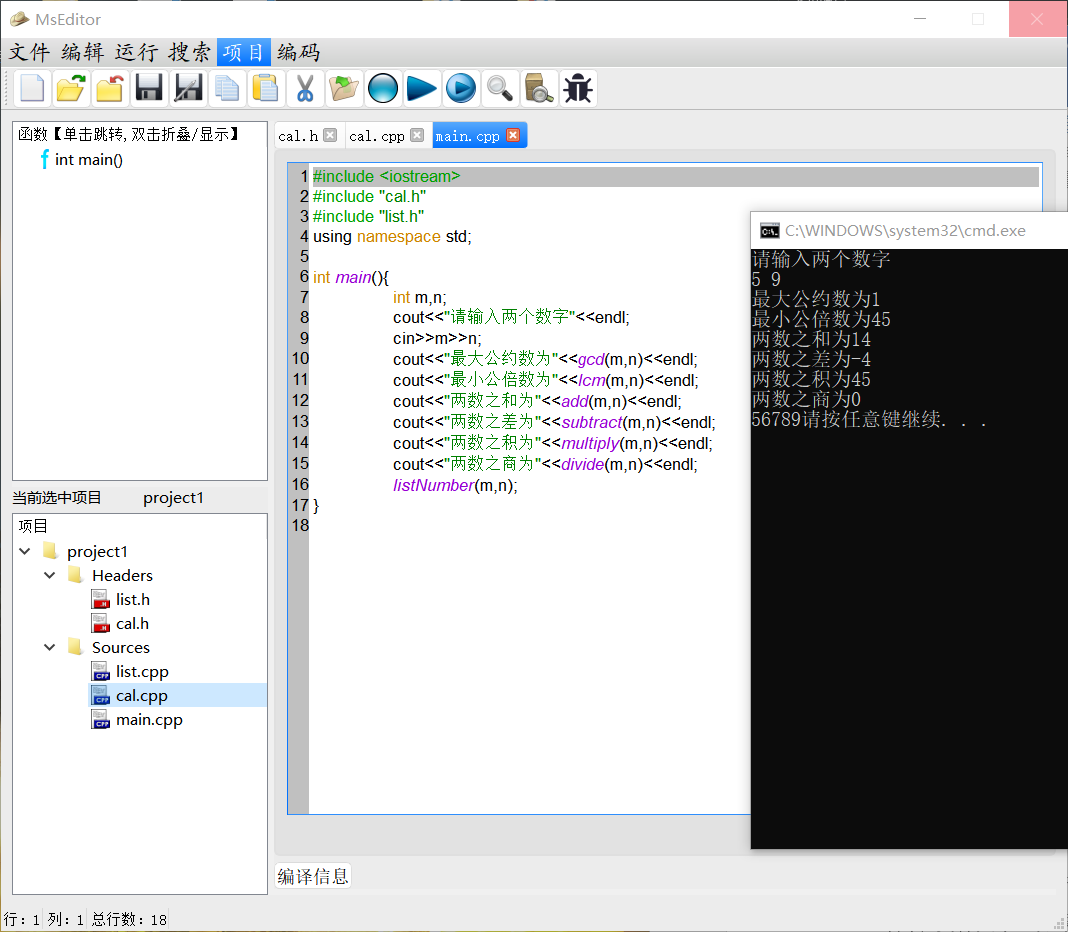
打开某个文件，即可以对文件进行编辑。可以同时打开多个文件，编辑了但是未保存的文件，对应的选项卡左侧会出现\*标记。

为满足代码编辑的特点，文本编辑框每一行都是无限延伸的，不会自动换行，光标停留行会有灰色高亮。编辑框下方有一个小跳转窗口，输入行数可以跳转到指定行。

左侧上方的函数栏显示当前文本编辑框所显示代码的所有函数，会随着页面的切换和代码的修改保存而刷新。单击函数栏中某个函数可以直接将光标跳转到该函数所在的位置，双击则可以将该函数折叠或是显示。

当使用菜单中“项目”下拉栏的“打开项目”功能打开某个文件夹的时候，项目栏会将该文件夹下面所有文件按照后缀筛选分类，分别显示在Headers与Sources中。使用“项目编译运行功能”可以对整个项目进行编译运行。

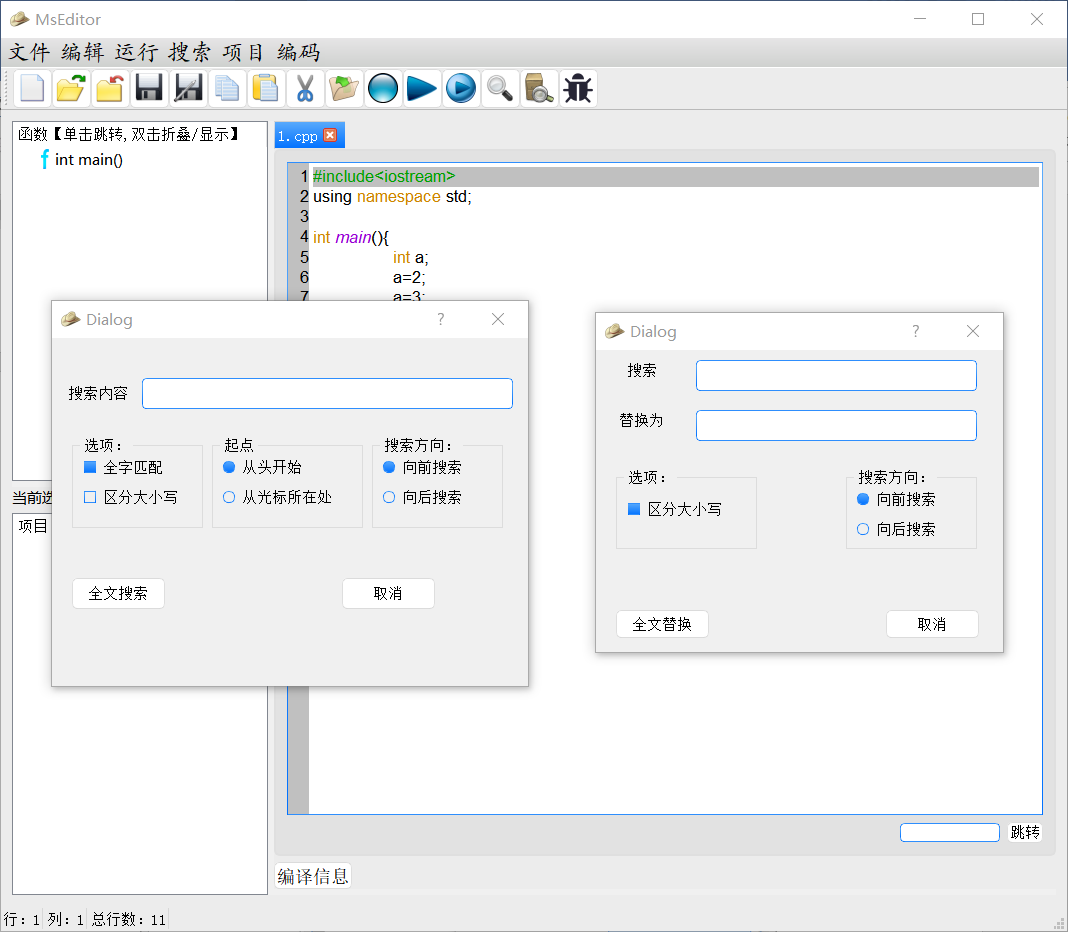
## 编译运行界面



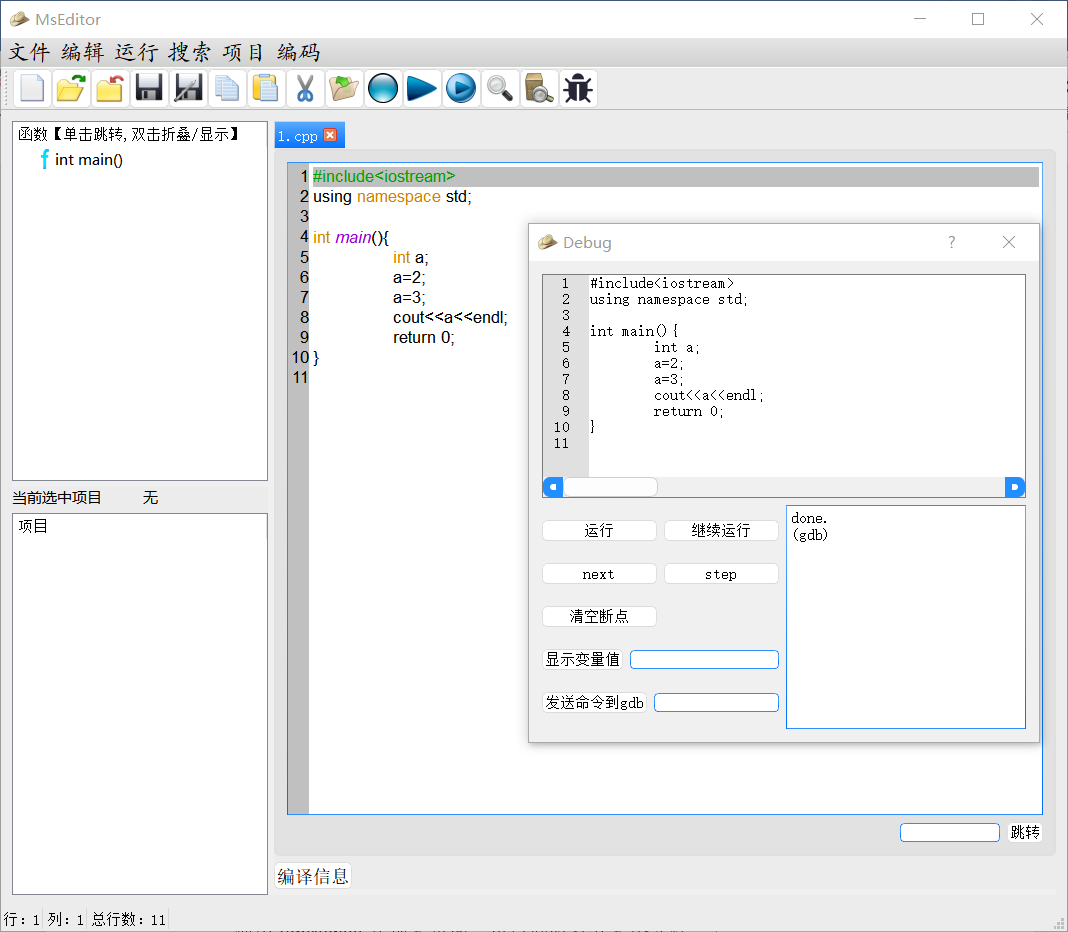
对项目或单个文件进行编译运行后（也可以先编译再运行），编译信息会自动显示在文本编辑栏下方。如果编译成功，会显示编译生成的文件名、最后修改时间和文件大小。如果编译失败，则会返回错误信息。

运行之后，会像dev一样出现单独的cmd窗口，可以在其中输入数据，观察程序的输出。

## 搜索替换界面



## Debug界面



# 功能及操作设计

## 多文件操作

使用tabWidget实现多页面，可以同时打开多个文件。

1. 新建文件

使用对话框获取要新建的文件路径，建立空文件，创建一个新页面。

1. 打开文件

使用对话框获取要打开的文件路径，将文件中的内容全部读出来，展示在一个新页面中。

1. 保存文件

将当前文本框的内容保存到相应文件中。文本框对应的选项卡左侧如果有\*,则代表这个文件经过修改后还没有保存。

1. 文件另存为

将当前文本框中的内容保存到新文件中。

1. 关闭文件

将当前文本框中的内容保存到文件中，然后关闭页面。

1. 复制、粘贴、剪切、撤销

直接调用QT自带的相应的功能函数，针对当前页面生效。

## 编译运行

用字符串拼接的方式组成编译运行命令，用system函数执行命令，对目标cpp文件进行编译和运行。如果编译失败，将编译后产生的错误信息输出重定向到文件中，最后读取到软件界面的编译信息框上显示；如果编译成功，则读取编译产生的文件的属性（路径、大小等）显示在编译信息框上。

运行之后，会像dev一样出现单独的cmd窗口，可以在其中输入数据，观察程序的输出。

## 项目操作

可以打开项目，软件会自动对项目文件夹下的文件进行分类显示。此时可以直接从项目栏中打开文件。同时支持对整个项目进行编译运行。

## 编码

软件支持打开和读写不同编码的文件，可以在菜单栏中的“编码”中选择UTF-8或者GBK。默认为GBK。

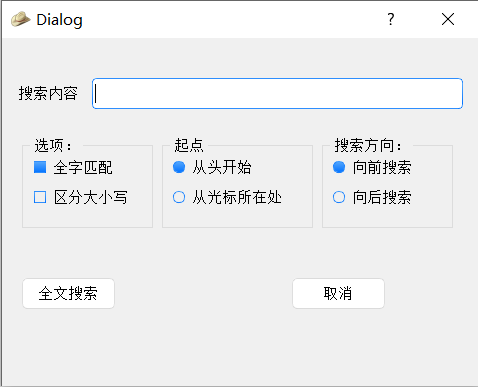
## 关键字高亮

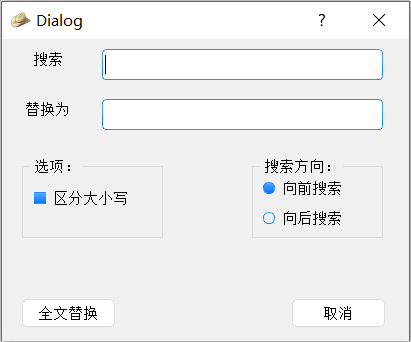
继承QSyntaxHighlighter类，使用正则表达式找出关键字、预处理命令、字符串、函数名、单行注释、多行注释，染上颜色。

## 函数操作

使用正则表达式找出函数头，记录其起始位置。将每个函数作为一项添加在函数栏中。单击函数栏某个函数可以使文本编辑框中的光标跳转到该函数起始位置，双击则可以将其折叠/显示。

## 搜索替换





针对当前打开的文本进行搜索替换。可选择不同的选项来进行操作，仅支持全文搜索以及全文替换。搜索使用自带函数find进行查找，然后改变光标至找到的文本处，并改变光标所选择文本的颜色。替换使用数据结构QStringList，将文本存入QStringList当中，对每行的String使用replace方法进行替换，然后重新settext到文本框中。

## 行号显示

自己写一个继承自QPlainTextEdit的类，然后在这个QPlainTextEdit左侧的页

边距加入一个widget，在文本框的文本改变事件触发时绘制widget用于显示行号。widget的宽度由数字字符的宽度来确定并及时改变。

1. 括号匹配

当用户将光标放置于括号前时，会将与该括号以及对应匹配的括号变色。

将文本框中的内容按行读入，遍历该行的每个字符，使用switch判断找到的

括号类型，如果为左括号，存入QList中，为右括号，则判断栈顶括号，匹配则栈顶出栈，并记录下这一对匹配的括号的位置信息。同时将所有找到的括号的位置以及它匹配的括号的位置存入另一个QList当中。当光标位置改变时，遍历这个QList，如果光标所在位置与该list中某一括号的位置符合，说明光标在括号上，则将该位置以及括号匹配的位置的括号改变颜色。

## 关键字自动补全

当用户在文本框中键入文本时，如在文件中有匹配的文本，会在光标下方显示一个窗口，展示所有的可用关键词，可以通过方向键来上下移动并使用Enter键来进行选择。

利用一个文件存储所有的关键字，然后自定义QCompleter来完成。

## 隐藏注释

处理键盘事件，当键盘按下CTRL+H的时候，将文本框中每行的内容转为

一个String，使用indexof寻找每行中可能的“//”以及“/\*”“\*/”。在还未找到“/\*”之前，优先寻找/\*，找到/\*后同时保证该行没有//，如有，则判断先后位置，//在前，/\*失效，继续寻找下一行；/\*在前，//失效。第一次找到有效的/\*后，将标记置为false，开始寻找第一个有效的\*/，同理，找到一个\*/后也要确认该行是否有//的存在。找到一个有效的\*/后，重新将标记置为true，然后继续寻找/\*。直至文本末尾。

隐藏注释后用户依然可以进行文本的编辑，编辑后再次打开注释，注释将仍然出现在相对位置正确的位置。

## Debug功能

这个功能是小学期最后一次展示之后添加的。用户点击debug按钮后弹出独立的界面，含有可以设置断点的文本框和一系列按钮和输入框，使用户很方便的进行debug。软件使用QProcess与gdb交互，实现打断点、运行、下一步、观察变量值、发送gbd命令等功能。