



华南师范大学

本科学生实验（实践）报告

院 系：计 算 机 学 院

实验课程：编译原理

实验项目：实验一 C++单词拼装起

指导老师：黄煜廉

开课时间：2023 ~ 2024 年度第 1 学期

专 业：计算机科学与技术

班 级：2021 级 1 班

学 生：李谋坤

学 号：20202831008

华南师范大学教务处

华南师范大学实验报告

学生姓名	李谋坤	学号	20202831008
专业	计算机科学与技术	年级、班级	2021 级 1 班
课程名称	编译原理	实验项目	C++单词拼装器
实验时间	2023 年 9 月 25 日		
实验指导老师	黄煜廉	实验评分	

一、实验内容

1. 把 C++源代码中的各类单词（记号）进行拼装分类。
2. C++语言包含了几种类型的单词（记号）：标识符，关键字，数（包括整数、浮点数），字符串、注释、特殊符号（分界符）和运算符等【详细的单词类别及拼装规则见另外的文件说明】。
3. 要求应用程序应为 Windows 界面。
4. 打开一个 C++源文件，列出所有可以拼装的单词（记号）。
5. 应该书写完善的软件设计文档。

二、实验目的

1. 了解和掌握词法分析器的工作原理；
2. 能够用简明的处理逻辑和数据结构对 C++中各类单词进行分类，并进行输出展示；
3. 能够使用图形化技术设计桌面应用程序来封装；

三、实验文档

将按软件工程规范书写的文档附加在这里。该部分应该有详细的实验项目分析、设计、实现及测试内容，例如系统的总体结构、数据结构的选择、关键算法的设计方案等。阐述时，应该尽量使用文字或图表的方式而不是简单地把实验项目的源代码粘贴进去，确保源代码的行数要比非源代码的行数要少得多。[在上交的报告中应把这些红色字删除]

3.1 实验分析阶段

基于实验的要求，该实验要实现的是在一段正常的 C++源码中，从头至尾挨个识别出各类单词，并进行获取和展示；如对如下代码：

```
//Test.cpp
#include<iostream.h>
main()
{
    int i;
    cin>>i;
    if (i>3) cout<<"ok";
    char c = 'a';
}
/* int a = 3*/
```

需要识别出其中“单行注释”、“与处理指令”、“标识符”、“数字”等等内容，

华南师范大学实验报告

学生姓名 李谋坤 学 号 20202831008
专 业 计算机科学与技术 年级、班级 2021 级 1 班
课程名称 编译原理 实验项目 C++单词拼装器
实验时间 2023 年 9 月 25 日
实验指导老师 黄煜廉 实验评分

最终输出如下：

1	#	特殊符号
2	Include	关键字
3	<iostream. h>	特殊符号
4	main	关键字
5	(特殊符号
6)	特殊符号
7	{	特殊符号
8	int	关键字
9	i	标识符
10	;	特殊符号

3.2 实验设计阶段

基于分析阶段的结果，可将整个实验的设计分为两部分：逻辑处理、界面展示。

3.2.1 逻辑处理部分

需要明确的是，我们需要逐行逐个字符进行获取和分析，判断目前这个字符是否可以单独成为一个单词，不行的话则要暂存，继续获取后续单词以至于出现能够将目前暂存字符标记为一个单词的情况。由于不能基于正则表达式的方式进行获取，因此采用 switch-case 内嵌 if-else 的形式作为主要的分流逻辑。

在此之前需要定义好单词的数据结构，以便能够存储其类型 ID、数值 val、符号 op 和内容 word：

```
// 单词结构
struct TokenStru
{
    TokenID ID;
    double val;
    char E[20]; // 存储数字E后的内容
    char op[3];
    char word[20];
};

TokenStru token; // 全局变量
char buffer[255];
int pos = 0; // 全局 index
```

● 主函数 String analyse(string str)

通过获取界面逐行扫描得到的一行文本，存入字符数组中 buffer，在全局变

华南师范大学实验报告

学生姓名	李谋坤	学 号	20202831008
专 业	计算机科学与技术	年级、班级	2021 级 1 班
课程名称	编译原理	实验项目	C++单词拼装器
实验时间	2023 年 9 月 25 日		
实验指导老师	黄煜廉	实验评分	

量 pos 下逐个字符进行扫描，即调用 `GetToken()` 函数；

`GetToken()` 函数将会将单个单词的属性存入 token 结构中。

返回主函数中将当前单词即判定结果存入字符串 ans 中，继续调用直到当前行的文本结束，并返回结果 ans；

```
string Analyse_1::analyse(string str) {
    pos = 0;
    string ans = ""; // 结果
    // 将str中的内容存入buffer数组
    strcpy(buffer, str.c_str());
    GetToken(); // 获取单词
    while (token.ID != ENDINPUT)
    {
        if (token.ID == NUMBER) {
            ans += std::to_string(token.val);
            if (token.E[0] != '\0') {
                ans += token.E;
            }
            ans += "\t数字";
        }
        else if (token.ID == ID) {
            if (KeyWords.find(token.word) != KeyWords.end()) {
                ans = ans + token.word + "\t关键字";
            }
            else if (preprocess_directive.find(token.word) !=
preprocess_directive.end()) {
                ans = ans + token.word + "\t预处理指令";
            }
            else ans = ans + token.word + "\t标识符";
        }
        else {
            if (special_symbols.find(token.op) != special_symbols.end())
            {
                ans = ans + token.op + "\t特殊符号";
            }
            else if (Operators.find(token.op) != Operators.end()) {
                ans = ans + token.op + "\t运算符";
            }
        }
    }
}
```

识别 token 是属于哪一种单词是由 ID 属性来决定，同时根据 ID 获取其 word

华南师范大学实验报告

学生姓名	李谋坤	学号	20202831008
专业	计算机科学与技术	年级、班级	2021 级 1 班
课程名称	编译原理	实验项目	C++单词拼装机
实验时间	2023 年 9 月 25 日		
实验指导老师	黄煜廉	实验评分	

或者 op。

● GetToken()函数

该函数的职责是对当前字符串对应的字符数组从前之后逐个扫描，如果累积字符可以满足某个单词的要求，那么将目前所有的字符取出，对应信息存入 token 中；

例如对于数字的识别，是要经历如下步骤的：

```
// 数字
if ( (buffer[pos] >= '0') && (buffer[pos] <= '9'))
{
    s = 0;
    while (buffer[pos] >= '0' && buffer[pos] <= '9')
    {
        s = s * 10 + buffer[pos] - '0';    pos++;
    }
    token.ID = NUMBER; token.val = s;
    if (buffer[pos] == '.')
    {
        int len = 0;
        s = 0; pos++;
        while (buffer[pos] >= '0' && buffer[pos] <= '9')
        {
            s = s * 10 + buffer[pos] - '0'; len++; pos++;
        }
        token.val = token.val + s / pow(10, len);
    }
    if (buffer[pos] == 'E' or buffer[pos] == 'e') {
        int len = 0;
        token.E[len++] = buffer[pos++];
        if (buffer[pos] == '+' || buffer[pos] == '-') {
            token.E[len++] = buffer[pos++];
        }
        while (buffer[pos] >= '0' && buffer[pos] <= '9') {
            token.E[len++] = buffer[pos++];
        }
        token.E[len] = '\0';
    }
}
```

识别 E 之前的部分，可以是整数也可以是小数，这部分的值可以由计算得

华南师范大学实验报告

学生姓名	李谋坤	学号	20202831008
专业	计算机科学与技术	年级、班级	2021级1班
课程名称	编译原理	实验项目	C++单词拼装机
实验时间	2023年9月25日		
实验指导老师	黄煜廉	实验评分	

出，但对于识别 E 后的值，这部分的值如果采用计算的方式，最终输出的内容会与原先的不符，于是决定对于这部分内容不计算而是采用字符数组直接存取；在结果输出处拼接在前半部分数值转字符串内容的后面。

再如对**运算符、特殊符号等**的识别，这部分总体采用 switch-case 语句作为分流逻辑；

```
// 运算符、特殊符号、注释、字符串
else if (buffer[pos] != '\0')
{
    int len = 0;
    switch (buffer[pos])
    {
        case '+': token.op[0] = buffer[pos]; // +
            if (buffer[pos + 1] == '=') { // +=
                pos++; token.ID = ADDEQ; token.op[1] = '='; token.op[2] = '\0'; break;
            }
            else if (buffer[pos + 1] == '+') { // ++
                pos++; token.ID = ADDADD; token.op[1] = '+'; token.op[2] = '\0'; break;
            }
            token.op[1] = '\0'; token.ID = ADD;
            break;
    }
}
```

例如若第一个字符是‘+’，那么就会有三种情况：‘+’、‘++’、‘+=’，需要直接在其后增设判断逻辑，如果某一情况符合直接存入对应 token 中，返回；

3.2.2 界面设计部分

界面设计的内容较为简单主要是完成两个文本框以及若干个按钮和标签的实现即可，最终的图形界面如下：



华南师范大学实验报告

学生姓名 李谋坤 学 号 20202831008
专 业 计算机科学与技术 年级、班级 2021 级 1 班
课程名称 编译原理 实验项目 C++单词拼装机
实验时间 2023 年 9 月 25 日
实验指导老师 黄煜廉 实验评分

● Main 函数

```
int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
    Widget w;

    w.show();
    return a.exec();
}
```

主要是创建了 QApplication a 以及布局 w，同时不断让 a 调用自身达到界面存在；

● 槽函数

```
void on_commitButton_clicked();
void on_cancelButton_clicked();
void on_resetButton_clicked();
void openButton_clicked();
```

四个按钮对应分析、取消、重置、打开文件，结合两个文本框 InputTextEdit、OutputTextEdit 来实现槽函数和按钮、文本框控件的链接；

```
/**
 * 提交按钮
 * @brief Widget::on_commitButton_clicked
 */
void Widget::on_commitButton_clicked()
{
    ui->OutputTextEdit->clear();
    // 获取 InputTextEdit
    QString text = ui->InputTextEdit->toPlainText();
    QStringList lines = text.split("\n");    // 分割
    // 遍历每一行并进行处理
    for(QString line : lines){
        qDebug().noquote() <<"原语句: " << line;
        std::string ans = aa.analyse(line.toStdString());    // 词法分析
        qDebug()<<"词法分析后内容是: "<<ans;
        QString qstr = QString::fromStdString(ans);
        ui->OutputTextEdit->append(qstr);    // 输出到输出文本框
    }
}
```

华南师范大学实验报告

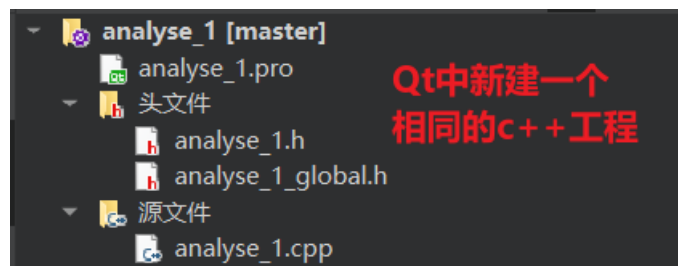
学生姓名 李谋坤 学 号 20202831008
专 业 计算机科学与技术 年级、班级 2021 级 1 班
课程名称 编译原理 实验项目 C++单词拼装机
实验时间 2023 年 9 月 25 日
实验指导老师 黄煜廉 实验评分

对于“分析”按钮的设计，它联系了输入文本框、外部导入的处理逻辑函数、输出文本框，而这些工作的开始是 click 即这个按钮被点击。

3.3 实验实现阶段

在上述两个部分的编码都实现之后，需要将两者合并在一起，由于逻辑处理部分实在 C++编译器中编写的，界面设计部分实在 Qt 项目中编写的，两者在语法上存在一点点差异，我并没有直接在 Qt 中来实现处理逻辑，而采用动态链接的方式，把逻辑处理部分当成一个外部库加入到 Qt 项目中，直接进行调用。

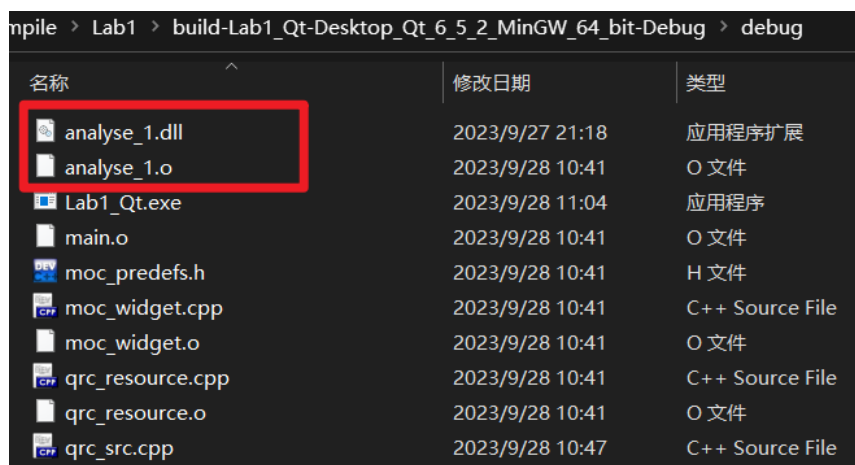
首先是在 C++项目（由于担心直接在 Visual Studio 中编译出来导入有些环境上不兼容，我在 Qt 中创建了一个完全相同 C++项目再进行编译）中对封装好的处理逻辑部分进行编译，需注意不能包含 main 函数，而是当成一个类来进行编写代码（这点很重要）；



其次在编译后的源程序文件夹中找到.dll 结尾的文件，这个就是动态链接库，其中包含了我们编写的函数；

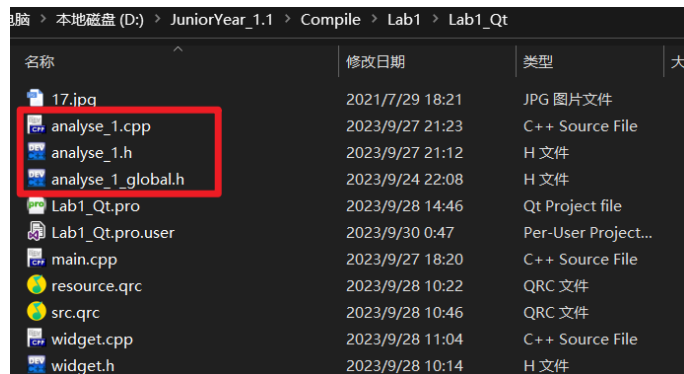


我们将其放入到 Qt 项目的源程序编译过后产生的文件夹中（和 exe 文件同一个目录），同时也要把 C++工程里的.h 文件和.cpp 文件放入源程序夹中；



华南师范大学实验报告

学生姓名 李谋坤 学号 20202831008
专业 计算机科学与技术 年级、班级 2021 级 1 班
课程名称 编译原理 实验项目 C++单词拼装器
实验时间 2023 年 9 月 25 日
实验指导老师 黄煜廉 实验评分

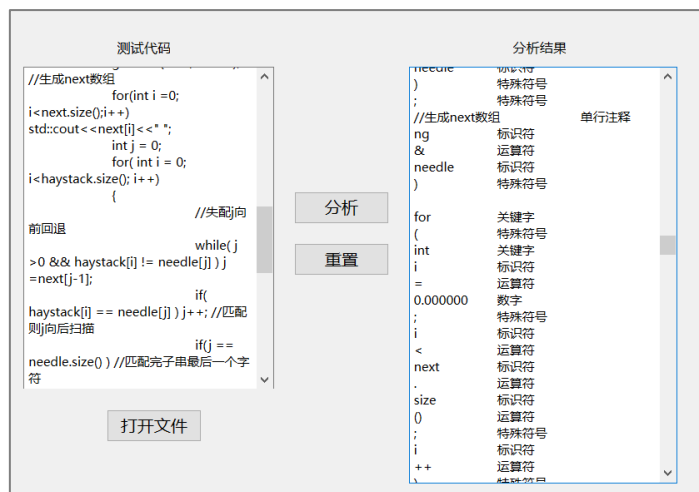


如此一来，在 Qt 项目中添加现有文件进入，完善好调用库的相关代码，即可实现整个项目的编写；

3.4 实验测试阶段

根据提交文件夹 03 的测试文件进行如下测试：

3.4.1 综合测试文件



华南师范大学实验报告

学生姓名 李谋坤 学 号 20202831008
专 业 计算机科学与技术 年级、班级 2021 级 1 班
课程名称 编译原理 实验项目 C++单词拼装机
实验时间 2023 年 9 月 25 日
实验指导老师 黄煜廉 实验评分

3.4.2 标识符、关键字

测试代码

```
//标识符
x y abc123 xy _xy xy_xy23

//保留字
if else do while for switch case break
continue default
cout
```

分析

重置

打开文件

分析结果

```
//标识符 单行注释
res 标识符
= 运算符
*<res<<std::endl; 字符串
d 标识符
:: 特殊符号
cout 关键字
<< 特殊符号
needle 标识符
[ 特殊符号
i 标识符
] 特殊符号
; 特殊符号

x 标识符
y 标识符
abc123 标识符
xy 标识符
_xy 标识符
xy_xy23 标识符

//保留字 单行注释
xy 标识符
```

测试代码

```
//标识符
x y abc123 xy _xy xy_xy23

//保留字
if else do while for switch case break
continue default
cout
```

分析

重置

打开文件

分析结果

```
abc123 标识符
xy 标识符
_xy 标识符
xy_xy23 标识符

//保留字 单行注释
xy 标识符
_xy 标识符
xy_xy23 标识符

if 关键字
else 关键字
do 关键字
while 关键字
for 关键字
switch 关键字
case 关键字
break 关键字
continue 关键字
default 关键字

cout 关键字
```

3.4.3 运算符、注释

测试代码

```
// 测试数据的组织：能够测试到每类单词或
每个单词

/*hjk*/
/=
//ajskldj
< << <= <
> >> >= >
= == =
* *= *
- - -= -- ->
+ ++ += +
! != ==
|| &&
```

分析

重置

打开文件

分析结果

```
/*hjk*/ 多行注释
/= 运算符
//ajskldj 单行注释

< 运算符
<< 特殊符号
<= 运算符
< 运算符

> 运算符
>> 特殊符号
>= 运算符
> 运算符

= 运算符
== 运算符
= 运算符

* 运算符
*= 运算符
* 运算符
```

华南师范大学实验报告

学生姓名 李谋坤 学 号 20202831008
专 业 计算机科学与技术 年级、班级 2021 级 1 班
课程名称 编译原理 实验项目 C++单词拼装机
实验时间 2023 年 9 月 25 日
实验指导老师 黄煜廉 实验评分

测试代码

```
// 测试数据的组织：能够测试到每类单词或  
// 每个单词  
  
/*hjkh*/  
/*=  
//ajskldj  
< << <= <  
> >> >= >  
= == =  
* *= *  
- - -= -- ->  
+ ++ += +  
! != ==  
|| &&
```

打开文件

分析结果

=	运算符
*	运算符
*=	运算符
*	运算符
-	运算符
-	运算符
-=	运算符
--	运算符
->	运算符
+	运算符
++	运算符
+=	运算符
+	运算符
!	运算符
!=	运算符
==	运算符
	运算符
&&	运算符

未实现部分：

1. 多行注释中当*/与*/不在同一行时*/前面注释内容的获取和识别；
2. Sizeof 运算；

3.4.4 特殊符号、串

测试代码

```
"sahjdak"  
'a'  
void Person::getName()  
{  
    return this->name;  
}  
  
char c[20] = {'\0'};  
  
cout<<"hello world";  
int a;  
cin>>a;
```

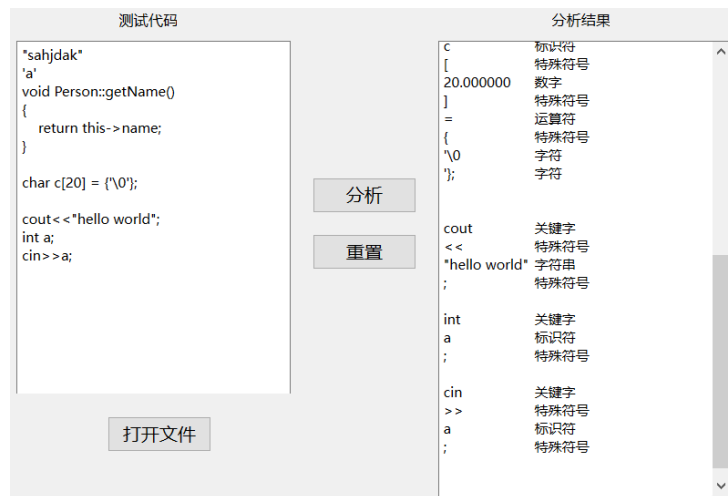
打开文件

分析结果

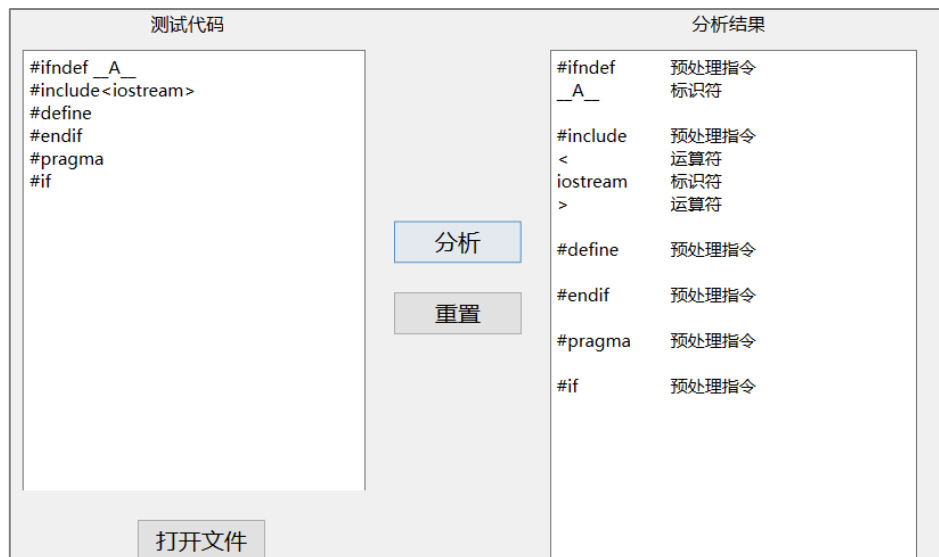
"sahjdak"	字符串
'a'	字符
void	关键字
Person	标识符
::	特殊符号
getName	标识符
()	运算符
{	特殊符号
return	关键字
this	关键字
->	运算符
name	标识符
;	特殊符号
}	特殊符号
char	关键字
c	标识符
[特殊符号
20.000000	数字

华南师范大学实验报告

学生姓名 李谋坤 学 号 20202831008
专 业 计算机科学与技术 年级、班级 2021 级 1 班
课程名称 编译原理 实验项目 C++单词拼装机
实验时间 2023 年 9 月 25 日
实验指导老师 黄煜廉 实验评分



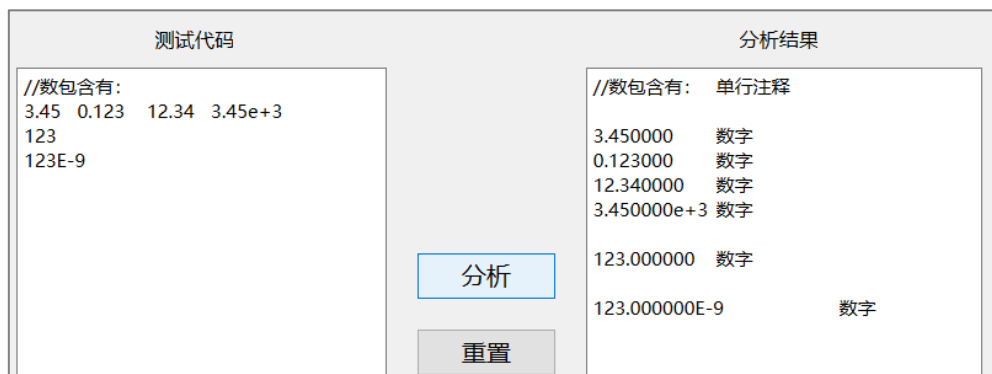
3.4.5 预处理指令



未实现部分:

1. #pragma once 的识别;

3.4.6 数字



未实现部分:

华南师范大学实验报告

学生姓名	李谋坤	学号	20202831008
专 业	计算机科学与技术	年级、班级	2021 级 1 班
课程名称	编译原理	实验项目	C++单词拼装器
实验时间	2023 年 9 月 25 日		
实验指导老师	黄煜廉	实验评分	

1. 0x 等进制符开头的数字；
2. 类似以 f、L 结尾的数字；

四、实验总结

本次实验总体而言难度较大，同时要学习新的软件、桌面应用程序的开发，所花费的时间也是不少的。

在完成了本实验之后，对词法分析器的运作机理有了一定的理解和掌握，对与 C++ 中单词的分类有了新的认识和理解；也学会了怎么使用 Qt 进行桌面应用程序的设计和开发；以及动态链接的相关机理和应用；

整体而言，本次实验的进展和结果相对成功，但尚有不足的地方：如一些功能并未设计得尽善尽美像数字的识别缺少对进制数的判别等；这些将在日后的实验中进一步完善。

五、参考文献

- [1] [QT 快速入门 | 最简单最简洁的 QT 入门教程 | 嵌入式 UI 哔哩哔哩 bilibili](#)
- [2] [Qt 之 qDebug\(\)打印和 QString 中文乱码 qDebug qstring-CSDN 博客](#)
- [3] https://www.bilibili.com/video/BV1m4411D7NG/?spm_id_from=333.999.0.0&vd_source=d825aca08d91fcae29a7839d2cabaadd （动态链接相关实现）