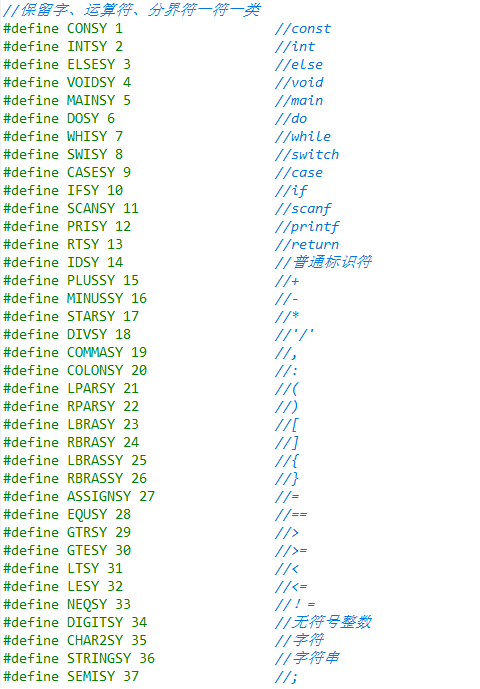
**词法分析程序实验报告**

1. **实验目标**

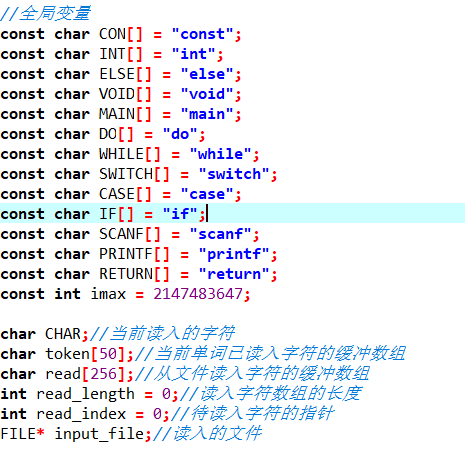
词法分析（lexical analysis）是计算机科学中将字符序列转换为单词（Token）序列的过程。进行词法分析的程序或者函数叫作词法分析器（Lexical analyzer，简称Lexer），也叫扫描器（Scanner）。词法分析器一般以函数的形式存在，供语法分析器调用。完成词法分析任务的程序称为词法分析程序或词法分析器或扫描器。本实验将编写**C0文法**的简单词法分析程序。

1. **实验要求**
2. 分析所给的C0文法，确定关键字，分界符，运算符，常量；
3. 对输入的代码段进行分析，输出每个单词的分类，并输出该单词的值，整数必须用二进制表示；
4. 常数默认为整数，必须区分分界符和运算符；
5. 运算符必须区分单字符运算符和双字符运算符；
6. 对不规范的单词有错误提示；
7. 要求文件输入；
8. 常数中可以识别小数。
9. **实验步骤**
10. **定义常量和全局变量**

为了实现“一符一类”，我们需要CO文法中所给的所有分界符、运算符、保留字定义成单独的分类，同时也将常量（无符号整数、字符、字符串）分别成类，并用数字标号作为类别码，方便后续实验的调用；



同时，我们遵循面向对象的思想，需要定义一些全局变量，方便之后多个函数的多次调用；

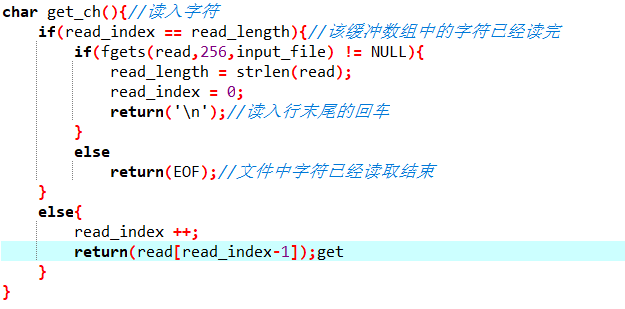


1. **成员函数的编写**

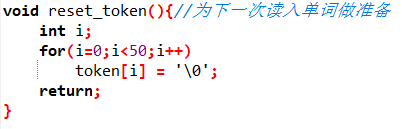
由于采用面向对象的编程，需要将各功能封装在函数中，通过函数间的调用来实现目标。

1. 主要功能函数

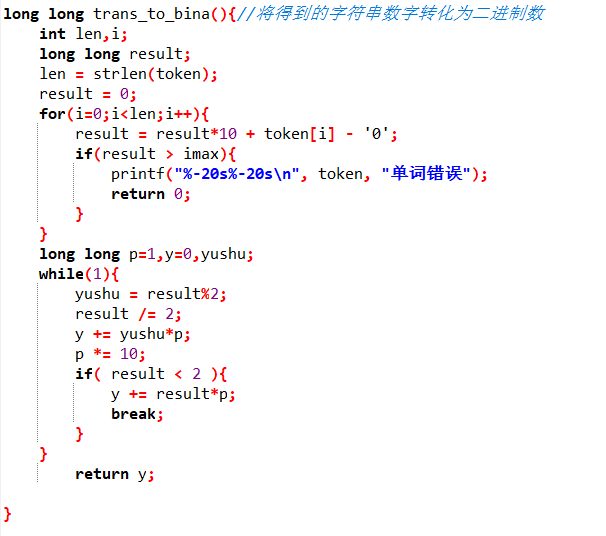
get\_ch()函数是用于将待分析字符读入全局变量CHAR中。其间需要判断文件输入缓冲数组read中的字符是否已经读完，如果读完，则将文件的下一行缓冲进read中，同时CHAR读入 \n 字符；如果没有读完，则CHAR 读入read的下一字符：



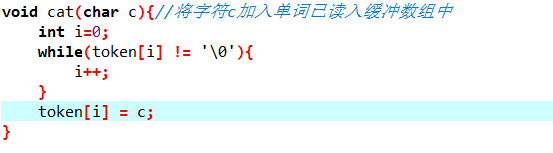
reset\_token()函数是将单词的字符缓冲数组token全置为 \0，为下一单词的字符缓冲做准备：



trans\_to\_bina()函数是将数字字符串转化为二进制的数字。首先将token数组保存的字符按照一定的算法转化成数字，再将该数字转化为二进制的数（实际上是十进制数，但其字符表示和token中的数字的二进制数相同）：

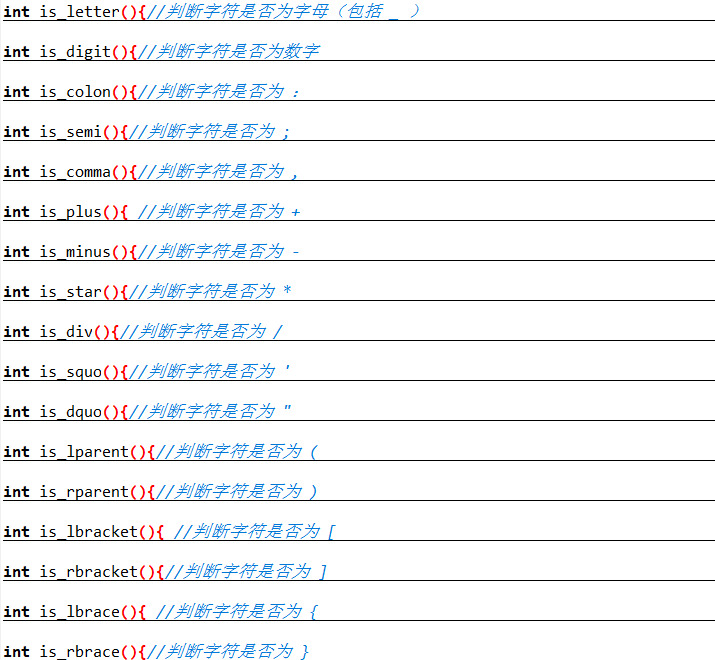


cat(char c)函数是将当前分析的字符CHAR连接到当前单词字符缓冲数组token后，即表明当前的CHAR属于当前单词：

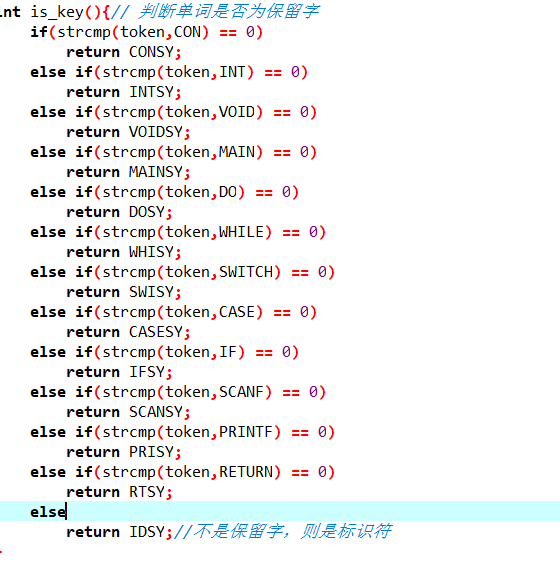


1. 判断函数

在对单个字符分析时，需要判断该字符是什么类型（字母、数字、某一符号等），所以对不同类型需要编写不同的判断函数（截图为部分）：



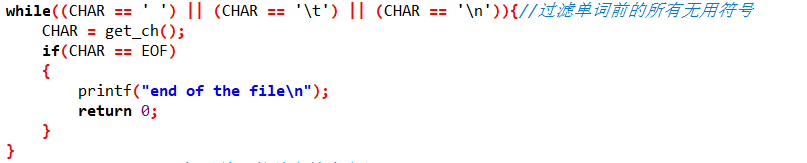
其中is\_key()函数则判断当前单词是否为某保留字，若不是，则当前单词为标识符：



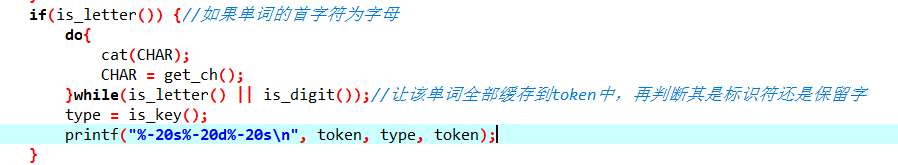
1. 单词分析函数

相关的调用函数编写完后，则需要对单个的单词进行分析。当然，在分析的开始并不知道单词的长度，这需要在不断的get\_ch()后进行判断。

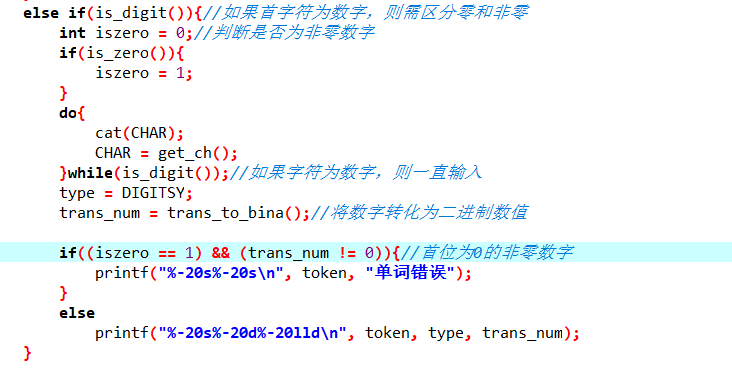
首先我们需要除尽单词开始前的空格、制表符和换行符：



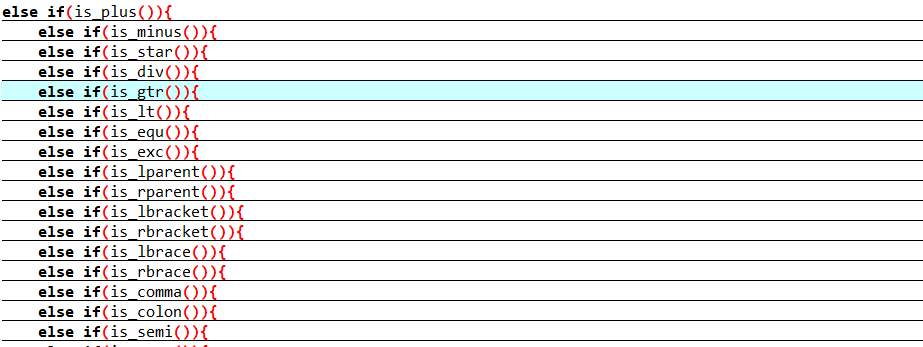
如果首字符为字母，则该单词为保留字或者标识符，单词将读到非字母非数字为止：



如果首字符为数字，则该单词只能为常数，单词读到非数字为止；如果首字符为0但单词值不为0，则出现单词错误，否则正常：



如果是其他合法的分界符或运算符等时，输出相应的单词、类型和值：



如果是单引号‘（双引号类似）,则需判断三种情况（**我的理解**）：

‘ 非法字符 字符：此时需要将 ‘非法字符 报错，同时read\_index回退一位（因为 字符 在下一个单词开始时需要重读）；

‘合法字符 字符：此时需要将 ‘ 报错，同时read\_index回退两位（因为 合法字符 字符 在下一个单词开始时都需要重读）；

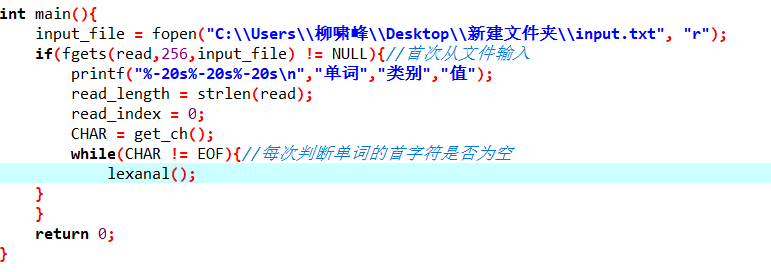
‘合法字符’：正常，该单词类型为字符类型；

‘非法字符’：整体报错：

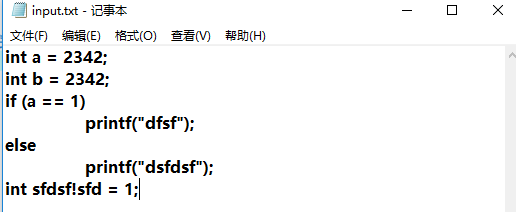


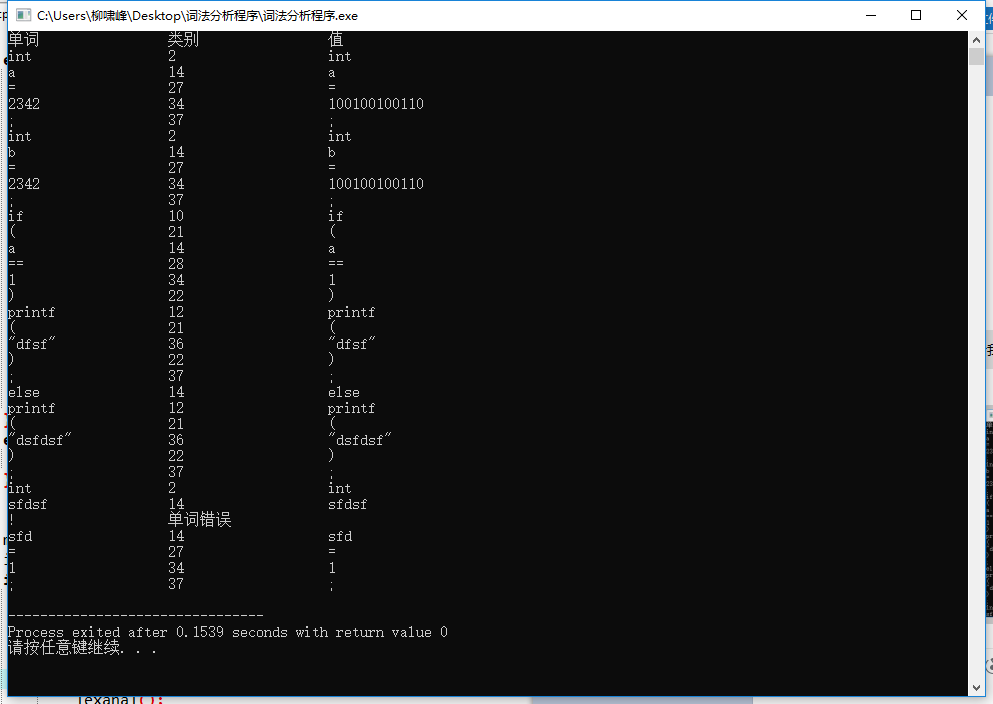
1. **主函数**

在主函数中，我们需要打开保存代码的文件，并将文件的第一行传入read函数中，并初始化相关全局变量，开始进行词法分析：



1. **运行结果**





1. **实验问题及解决：**
2. **关于字符及字符串报错问题？**

解决：由于C0文法中并未将单引号和双引号作为分界符，但作为字符和字符串的必要因素，当出现前后不匹配时，应该怎样报错？该将引号与字符作为整体报错还是将引号剥离出来单独报错？这一点我不是很清楚，我的解决办法见实验步骤。

1. **实验感想**

我觉得词法分析程序的编写应该要注意以下几点：

首先要注意字符读入的时机。可以把每个输入到CHAR中的字符相成有两个状态：分析过和未分析。如果CHAR中的字符被连接到了token上或者被报错，则该字符算作分析过，则需要get\_ch()读入新字符到CHAR中；否则该字符未分析，不需要读入新字符。当然主函数第一次调用单词分析子函数lexanal前需要初始化并读入第一个字符。

其次应该注意判断单词类型的完备性。由于单词并不是直接分割好的，需要在读入字符的同时判断该单词是否已经结束。所以在考虑怎样的字符搭配能得到怎样的单词类型时一定要严谨、完整（这就为什么在单词分析阶段我写了200+的代码）。

最后要注意代码的规范性和兼容性。由于最终要做一个完整的编译器，之后的工作如语法分析程序等都需要调用到词法分析程序，所以将该程序的各种功能封装好，类型定义等尽量规范，这样能够为后来的编程带来一定的方便和帮助。