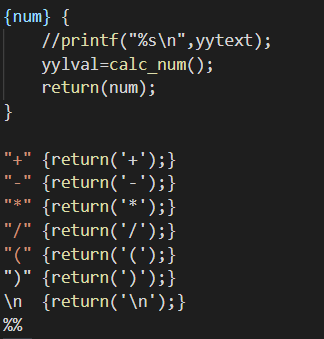
使用YACC实现语法分析

YACC简介

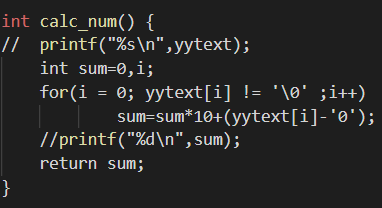
YACC是一款语法分析器的自动生成工具。用户按照YACC规定的规则写出文法说明文件，该文件一般以.y为扩展名，YACC编译器将此文法说明文件转换成用C编写的语法分析器文件filename.tab.c以及对应的头文件 filename.tab.h。

在使用YACC编写语法分析器之前，我们需要在该文件夹中使用Lex完成文法分析的编写，这个文件里应该包含语法分析驱动函数yyparse()以及LALR分析表。语法分析驱动程序则调用yylex()这个函数获取输入记号，每次调用yylex()都能获取一个输入记号。

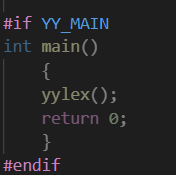
Lex程序内容分析



Lex程序中主要进行数字以及计算符号的定义。



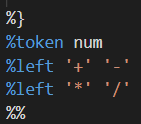
同时，Lex中定义了calc\_num()函数，负责将识别到的数字转换成其对应的数值。



在有lex程序编译成的c语言程序中，含有yylex()函数被语法分析程序的驱动程序调用。

YACC程序内容分析

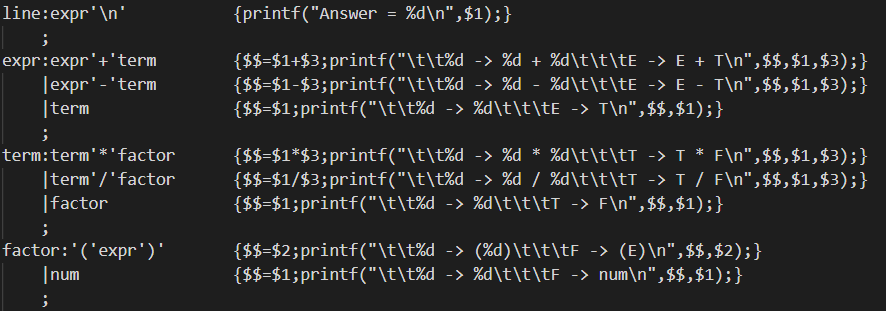
在YACC程序中首先定义了文法中使用的终结符。以%taken开头定义的终结符不具有优先级关系，以%left开头定义的终结符具有优先级关系，先定义的终结符优先级较高，后定义的终结符优先级较低，left表示符号满足左结合规律。



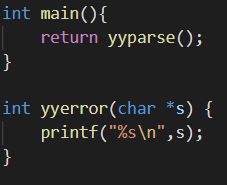
接下来是规则段，记录了文法的产生式以及规约整个产生式时应该采取的动作。

产生式中的每一个 | 分隔的部分都单独占一行，整个产生式写完后，结尾处应加上分号。

由下图的代码可见，在产生式规约时，程序分别执行了数值的计算以及将推导过程输出两个动作。



最终在YACC代码中定义了程序入口函数以及遇到错误时的错误显示函数。



编译及运行过程

在windows环境下，lex程序由flex程序替代，YACC程序由Bison程序替代，分别使用两个程序将源代码编译成c语言代码。

|  |
| --- |
| bison -d C:\Users\Anthony\Desktop\compiler\compiler\src\yyac\second.y  flex -d C:\Users\Anthony\Desktop\compiler\compiler\src\yyac\first.l |

接着将c语言代码进行联合编译

|  |
| --- |
| gcc -o mycc C:\Users\Anthony\Desktop\compiler\compiler\src\yyac\second.tab.c C:\Users\Anthony\Desktop\compiler\compiler\src\yyac\lex.yy.c |

最终得到可执行程序，能够正常运行：

