

# 编译实习小作业 基于语法制导翻译的简易计算器

---

罗昊 1700010686

## 简介

该计算器基于基于语法制导的翻译,由 flex 和 Gnu Bison创建,支持基本四则运算和赋值运算.

四则运算支持括号和优先级,支持整数运算和浮点数运算,除法运算规则与C语言相同.

支持通过赋初值的方式创建变量,变量名与c语言标识符风格相同,不支持中文变量名.

支持等号串联赋值,类似c语言,例如a=b=3.

支持多语句串联,可用分号或回车作为分隔符.

遇到非法字符,使用未定义的变量,语法错误时会给出相应的错误提示,可继续输入新语句.

退出命令为quit

## 编译

依赖 flex bison g++

测试环境为 flex-2.6.4 , bison-3.0.4 , g++-7.3.0

已写makefile, 直接make即可编译, 输出可执行文件为calc.

```
make
./calc
```

## 示例

```
1+2* (1+3)
9

2.0 * 3
6.000000

3/ 2
1

3./2
1.500000

a=3*-1
```

```

-3

a+5
2

b+3
Error encountered: identifier b not defined!

b=a=13.1
13.100000

c=2; c+1
2
3

编译实习=100
Error encountered: Illegal Character!

2+
Error encountered: syntax error

quit

```

## 代码结构

代码由两个文件组成:calc.l和calc.y

calc.l

由三部分组成:#include 部分,词法规则,附加代码.

第一部分包含了bison生成的y.tab.h,其中有一些类型的定义和token宏的定义.

第二部分为词法规则,主要的token有运算符、整数、浮点数、标识符、分隔符、quit以及含有非法字符的语句.

第三部分为附加代码,只有yywrap()函数.

calc.y

由三部分组成:类型定义、token声明和附加声明和定义,语法规则,附加代码

第一部分有

%code requires部分会被附加到y.tab.h中,包含需要include的库,自定义的结构体,yy系列函数的声明. (%code requires的作用查阅自[Gnu Bison文档](#))

%union部分,定义YYSTYPE为union类型. (%union的作用查阅自[Gnu Bison文档](#))

%code部分,包含一个全局变量和所需函数的定义.

全局变量为一个作为符号表的unordered\_map,键类型为string,值类型为一个包含type和value的结构体,其中type代表变量是实数还是整数,value是一个union,储存变量的值.

所需函数有用于进行单步运算的calc函数和negative函数,以及用于报错的yyerror函数.

然后是token的声明和类型定义,以及部分非终结符号的类型定义.

接下来%right 和 %left 声明运算符的结合性和优先级顺序 (%right %left 和 %prec 的作用和优先顺序查阅自[Gnu Bison文档](#)

第二部分是语法定义,设置了三个非终结符,stmts expr 和 assignexpr.

stmts是多条语句 expr是单条非赋值语句 assignexpr是单条赋值语句.

第三部分为附加代码,包含main函数.