文法：

Expression -> expression + term

Expression -> expression - term

Expression -> term

Term -> term\*factor

Term -> term/factor

Term -> factor

Factor -> (expression)

Factor ->id

程序说明：

本程序实现简单的四则运算计算器，特点如下：

1、支持整型和浮点型的运算，也支持整型和浮点型相互转换

2、可以识别的运算符有：+、-、\*、/、(、)、=

3、用户需要先声明要使用的变量，使用关键词float或者int

4、变量须初始化后才能使用，使用未赋值的变量会报错

5、用户的每句输入以“;”结尾，最后以“.”结束全部输入

6、变量名可以是由数字和字母组成，但首字符必须是字母

7、输出语句使用write(a)，输出并换行，其中a为int类型或者float类型的变量名

8、本程序也实现了对未声明变量、未初始化变量、除0和句法错误的报错，程序会指出出错的行数，并退出。

句法错误的说明：声明变量的格式固定为：int/float 变量名；

一次只能声明一个变量

Write()函数的格式固定为：write(变量名)

不允许在一句话中连续使用等号赋值。

算术表达式的错误也称为句法错误

数据结构说明：

1、 typedef struct token {

int TokenType;

char value[32];

}Token;

用于存储词法分析得到的单词，value是这个单词的值（abc的value即“abc”， 123的value即“123”），因为长度为32，所以本程序所有数字和变量名不超过32字节。TokenType是这个单词的类型，具体定义为：’+’ = 1, ‘-’ = 2, ‘\*’ = 3, ‘/’ = 4, ‘(‘ = 5, ‘)’ = 6, 整数 = 7, 浮点数 = 8, 变量 = 9, 关键词 = 10, ‘=’ = 11, ‘;’ = 12。

2、typedef struct var {

int type = -1;

char name[32] = {};

int int\_val = 'a';

float float\_val = 'a';

}Var;

用于存储变量，name即变量名，type是变量类型，定义同上。如果是浮点变量，就将变量的值储存在float\_val，如果是整型变量，就将变量值储存在int\_val。未初始化的变量float\_val和int\_val都是初值’a’。

函数说明：

1. int namecmp(char\* a, char\* b)

用于比较两个长度为32的数组的内容。两数组完全相同返回1，否则返回0。

1. int opcmp(Token\* TokenList, int op)

返回运算符的优先级，’+’、’-’为4，’\*’、’/’为5。

1. int GetInput(char\* input, char\* file)

按字节读取输入文件，将每个字节存入input数组，最后将输入结束的标识’.’改为’;’，并返回读取的字节数。

1. int Scaner(char\* input, int length, Token\* TokenList)

根据input数组识别单词，生成Token类型的单词并存入TokenList数组，最后返回生成的单词数。

1. int Reader(Token\* TokenList, int length, Var\* VarList, int\* CMDList)

根据TokenList生成Var类型的变量，并存入VarList数组。同时，将每句开头的token在TokenList中的下标存入CMDList，这就分出了句子。

1. int calculator(char\* var, int start, int end, Token\* TokenList, Var\* VarList, int VarNum)

计算只有数字和运算符的表达式。成功计算返回0，并会更新变量表，表达式有语法错误返回1。

1. void calc(Var\* VarList, int\* CMDList, int VarNum, Token\* TokenList, int TokenLength)

根据CMDList逐句读取，只处理是算术表达式的句子。先判断式中等号的左值是否已声明，未声明则报错。然后将式中的变量转化为数字，若变量未初始化则报错。然后调用calculator()函数计算当前算术式。

1. void Output(Token\* TokenList, Var\* VarList, int TokenLength, int VarNum, int\* CMDList)

根据CMDList逐句读取，只处理write()语句，根据变量声明的类型输出该变量的值。

运行样例：

1. float a;

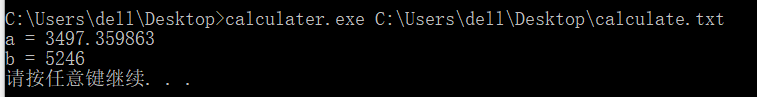
int b;

a =(10.44\*356+1.28)/2+1024\*1.6;

b=a\*2-a/2;

write(a);

write(b).



正常运行。

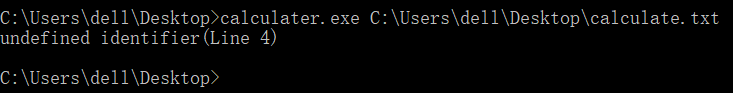
1. float a;

int b;

a = (10.44\*356+1.28) / 2 + 1024 \* 1.6;

b = a \* 2 - c/2;

write(b).



第4行c未定义，报错退出。