**简单语义分析扩展实验**

**一、实验目的**

1. 掌握单遍扫描编译程序的组织方法；
2. 掌握符号表的基本管理方法；
3. 掌握中间代码生成的一般方法。

**二、实验内容**

针对下面的文法

**start→ DS.**

**D→B; D**

**D→ε**

**B→int L | real L**

**L→id | L，id**

**S→ V := E H**

**H→；S | ε**

**E→TR**

**R→+TR |-TR |ε**

**T→FP**

**P→\*FP |/FP |ε**

**F→( E ) | id**

**V→id**

其中id是字母开头后面是数字后者字母的字符串。

完成词法、语法、语义分析的编译器前端。具体要求包括：

1. **实现算术运算+、-、\*、/运算，运算满足左结合，且满足\*，/优先级别高于+、-；将合法的输入翻译成三地址代码：**

例如输入

real a,b;

int c,d;

c:=c+d\*b;

d:=b+a.

则输出

(\*,d,b,T101)

(+,c,T101,T102)

(:=,T102,,c)

(+,b,a,T103)

(:=,T103,,d)

1. **类型检查：**

**即对所有的运算包括赋值，需要考虑运算对象的数据类型是否一致，不一致报告语义错。**

1. **禁止同名标识符的重复申明；**

**三、提交实验**

**提交内容**：

1. 实验报告：解释你的设计方案；
2. 编译通过的程序代码。

**四、实验参考**

对如下文法：

**P→ DS.**

**D→B; D | ε**

**B→int L | real L**

**L→id | L，id**

**S→ V := E H**

**H→；S | ε**

**E→TR**

**R→+TR | ε**

**T→( E ) | id**

**V→id**

提供实现如下功能的C语言版本的编译器前端“语义分析.cpp”和“语义分析.h”。该程序的具体功能包括：

1. **词法分析；**
2. **语法分析；**
3. **语义功能包括：**
4. **非终结符号D实现定义两种类型int, real变量的声明；**
5. **非终结符号S实现变量之间的+，:=(赋值运算)**
6. **两个关键字int和real**
7. **变量之间的+，: =(赋值)运算只能使用声明过的变量，所以检查使用的变量是否声明过。**
8. **对每个+，: =(赋值) 运算生成一条四元式如（+，A，B，T1），其中T1是临时变量**

例如，输入

real a,b;

int c,d;

c:=c+d+b;

d:=b+a.

则输出

(+,c,d,T101)

(+,T101,b,T102)

(:=,T102,,c)

(+,b,a,T103)

(:=,T103,,d)

例如输入

real a,b;

int c,d;

f:=c+d+b;

d:=b+a.

则语义分析要能识别没有申明使用标识符错

**五、实验提示**

1. 所提供的源代码仅供参考，给大家直观展示实现表达式的语义分析实现。所给的程序采用标准的递归下降分析方法实现。大家可以根据任务和自己的设计对文法进行等价变换，或者对程序结构进行调整（例如循环代替递归）。
2. **算术运算+、-、\*、/运算，运算满足左结合，且满足\*，/优先级别高于+、-可以由上下文无关文法定义。**