姓名： 黄珀芝 学号：20201050331 专业：计算机科学与技术

**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**

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；**
2. **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；**
3. **禁止同名标识符的重复申明；**
4. **实验分析与设计**
5. 定义以下（翻译文法）：

|  |
| --- |
| start→ DS. D→ B; D D→ε B→ int L { L.type := int } | real L { L.type := real } L→ id { A.type := L.type enter(v.entry, L.type)} A A→ ,idA { A1.type := A.type enter(v.entry,A.type)} A→ε S→ V := E { gen( ":=", E.place,0,V.place) } H {if E.type != H.type error\_type; else V.type=E.type} H→;S | ε E→T { R.place:=T.place} R {E.place:=R.place; if T.type != R.type error\_type; else E.type=T.type} R→+T { R1.place:= newtemp; gen( "+", R.place, T.place , R1.place) } R {R.place=R1.place; R.type=R1.type} R→-T { R1.place:= newtemp; gen( "-", R.place, T.place , R1.place) } R {R.place:=R1.place; R.type=R1.type} R→ ε {R.place=R.place; R.type=R.type} T→F { P.place:=F.place} P { T.place:=P.place; if F.type != P.type error\_type; else T.type=F.type} P→\* F { P1.place:= newtemp; gen( "\*", P.place, F.place ,P1.place) } P{P.place:= P1.place;P.type=P1.type } P—>/ F { P1.i:= newtemp; gen( "/", P.place, F.place , P1.place) } P {P.place:= P1.place;P.type=P1.type } P→ε {P.place=P.place; P.type=P.type} F→( E ) { F.place := E.place; F.type=E.type} F→id {F.place := found(id).place; F.type=found(id).type} V→id {V.place:=found(id).place; V.type=found(id).type} |

1. 按照实验要求对结构体的改写及函数内容的增加：
2. 实现算术运算+、-、\*、/运算：

答：在结构体内加入minus（-）、addtoken（+）、times（\*）、divide（\），加减乘除运算，运算满足左结合，且满足乘除优先级别高于加减。

1. 增加乘除运算，添加函数P(info Pi)，分别处理乘除运算。

用代码来表示：

|  |
| --- |
| 乘法运算： gen(times, Pi.place, F\_info.place, tv);  P\_info = P(F\_info); |
| 除法运算：gen(divide, Pi.place, F\_info.place, tv);  P\_info = P(F\_info); |

3、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。

实现赋值类型检查：

答：在S（）函数中，得到V（）返回值中的类型信息以及E（）返回值中的类型信息，将其进行比较，如果二者类型不一致，则说明出现了类型赋值类型不一致的情况，进行报错。

实现运算类型检查：

答：P（）、E（）、T（）、R（）函数中，增加类型判断，在T（）中，由于T->FP，则需要将F以及P的类型进行检查。

1. 禁止同名标识符的重复申明。

为了进行重复定义检查，在插入符号表时，需要进行查询，如果在符号表中出现过当前变量，则说明出现了重复定义。

|  |
| --- |
| //检测重复定义  for (int i = 1; i < tx; i++) {  if (strcmp(table[i].name, id) == 0) {  printf("重复定义: %s\n", id);  exit(0);  }  } |

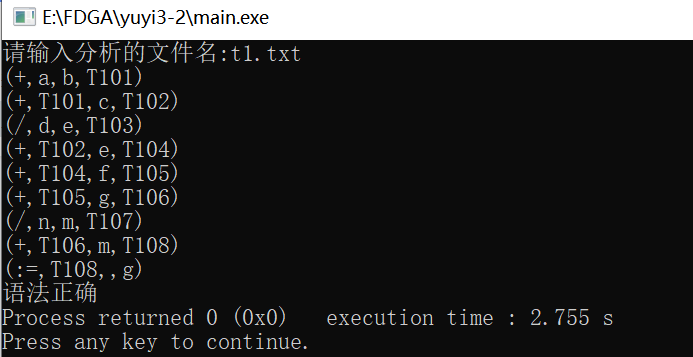
1. **实现与结果分析**

将所给的文档t1-t4.txt输入：

①输入：

real a,b,c,d,e,f,g,n,m;

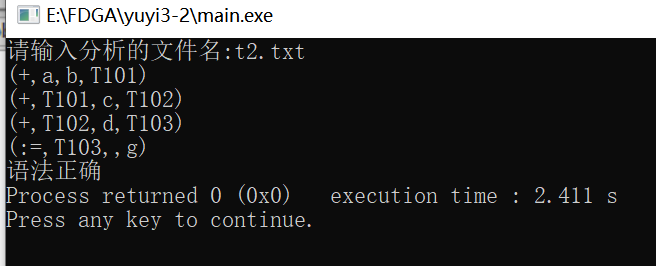
g:=a+b+c+d/e+f+g+n/m.



②输入：

real a,b,c,d,e,f,g,n,m;

g:=a+b+c+d.



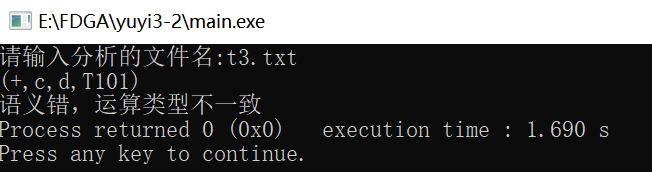
③输入：

real a,b;

int c,d;

c:=c+d+b;

d:=b+a.



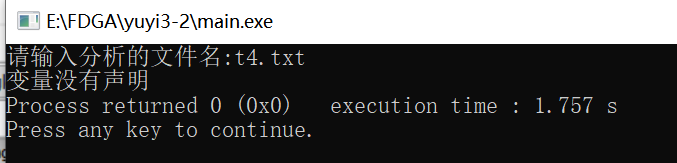
④输入：

real a,b;

int c,d;

f:=c+d+b;

d:=b+a.



程序可以对正确程序做出正确的返回，能够对运算类型错误、赋值两端类型不一致错误、重复定义错误、变量未声明错误等进行正确的检查，完成了实验要求，符合语法规则。