**语法分析、语义分析、符号表、中间代码测试**

目录

[1.测试程序清单 1](#_Toc20971)

[2.覆盖情况： 2](#_Toc15190)

[3.部分测试结果： 3](#_Toc5296)

[3.1语法分析结果： 3](#_Toc1596)

[3.2符号表： 3](#_Toc15740)

[3.3函数表： 4](#_Toc9728)

[3.4中间代码： 4](#_Toc27606)

[3.5优化后的中间代码： 5](#_Toc23967)

在递归下降过程中进行代码生成，因此，只有当错误的数量为0的时候，生成的中间代码才会是正确的。

在调试过程中利用中间代码检查逻辑是否正确。

对于递归下降的过程，通过分析语法分析的错误提示信息（调用error.cpp模块），打印出的符号表以及中间代码进行测试。

# 1.测试程序清单

|  |
| --- |
| 1\_func\_declare.txt |
| 1\_vardef.txt |
| 1\_variable\_rename.txt |
| 2\_array.txt |
| 2\_expression.txt |
| 2\_return\_value.txt |
| 3\_function\_call.txt |
| 3\_function\_call\_2.txt |
| 3\_judge.txt |
| 3\_while.txt |
| 4\_array.txt |

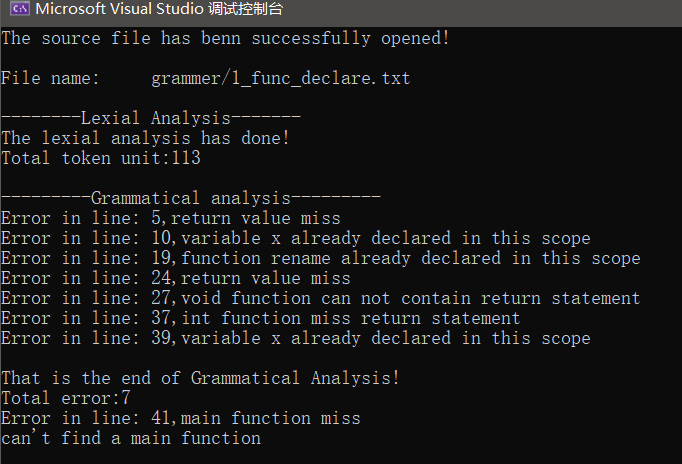
# 2.覆盖情况：

|  |  |  |
| --- | --- | --- |
| 功能 | 语义检查点 | 是否通过 |
| 函数定义 | 返回值检查 | 通过 |
| 函数重命名 | 通过 |
| 返回值 | 通过 |
| main函数声明和return | 通过 |
| 参数与局部变量命名 | 通过 |
| 变量和常量定义 | 变量重命名 | 通过 |
| 常量未初始化 | 通过 |
| 初始化错误 | 通过 |
| 表达式 | 表达式优先级 | 通过 |
| 函数调用 | 参数传递，型参与实参匹配 | 通过 |
| 返回值 | 通过 |
| 条件判断 | if语句 | 通过 |
| while语句 | 通过 |
| 数组 | 数组下标检查 | 通过 |

# 3.部分测试结果：

## 3.1语法分析结果：

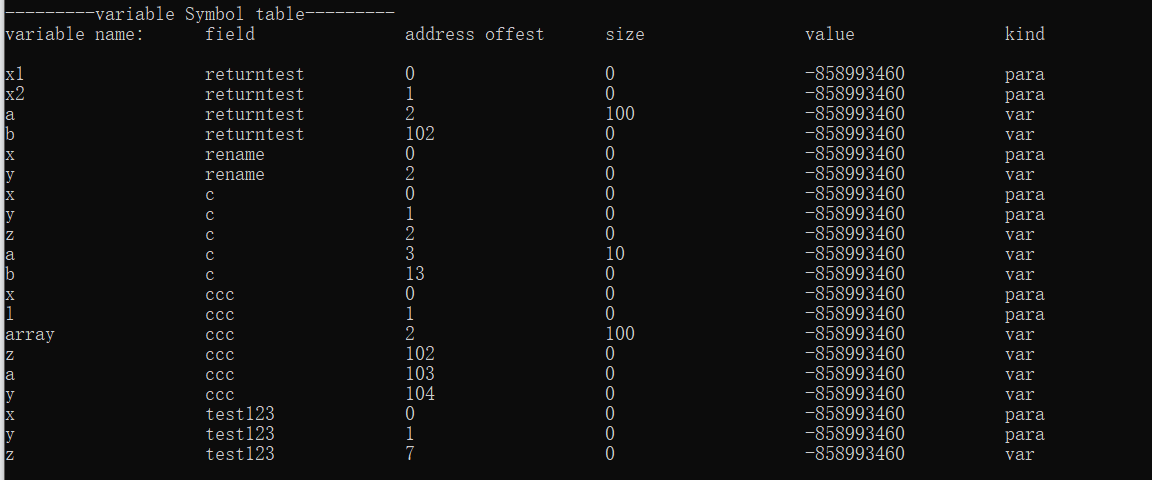
语法分析采用递归下降的方式分析词法分析得到的token序列，对于不符合sysy语义的语句，调用错误处理模块，输出错误类型和行数。



## 3.2符号表：

符号表包括了函数表和变量表。

变量表的内容：变量名，作用域，地址偏移，大小（基本类型大下为0），初值（未经初始化的变量显示一个随机值，常量初值必须定义），类型



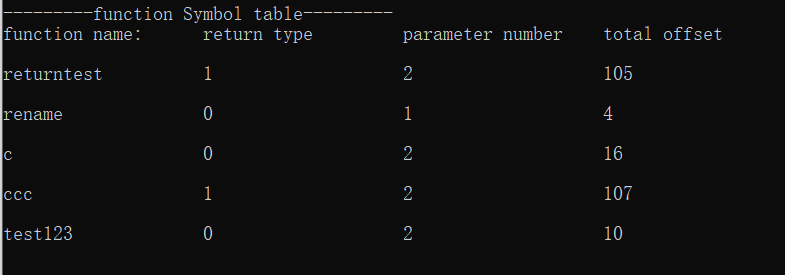
根据sysy的文法，变量值可以

VarDef 用于定义变量。当不含有‘=’和初始值时，其运行时实际初值未定义。

未经初始化的变量会随机赋值，因此在符号表打印时会出现一个随机数。

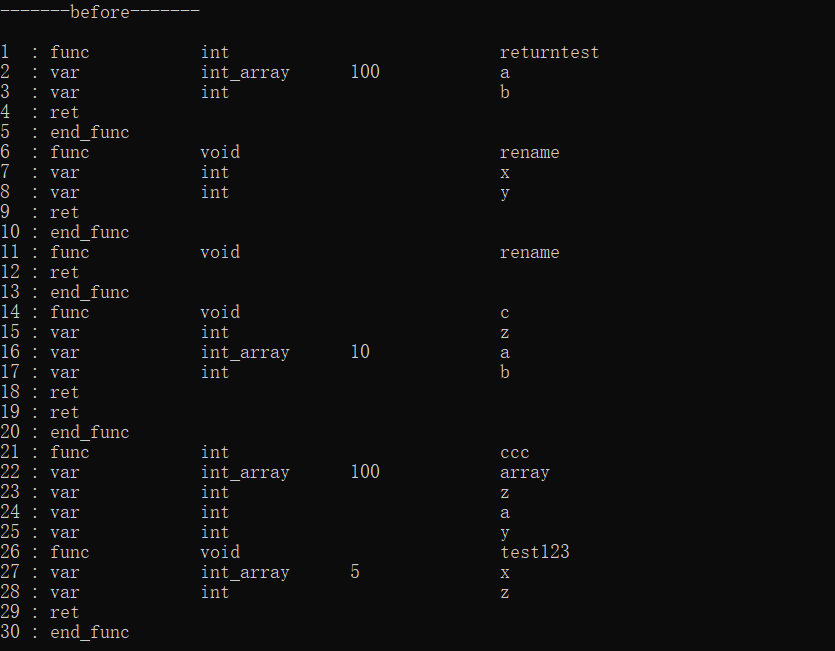
## 3.3函数表：

包括的信息有函数名，返回类型，参数个数，总偏移量（参数个数+临时变量的大小）



## 3.4中间代码：

在递归下降过程中生成中间代码序列



## 3.5优化后的中间代码：

经过opt模块对中间代码进行优化，得到优化后的中间代码。

