

****

软 件 学 院

《编译技术》大作业报告

**题　　目 PL/0编译器的设计与实现**

**姓　　名 陈澄**

**学　　号 32420212202930**

**班　　级 软工三班**

**实验时间 2024/6/6**

**2024 年 6 月 6 日**

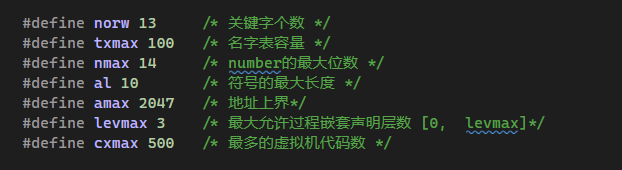
# 实验要求

设计并实现一个PL/0语言的编译器，能够将PL/0语言翻译成P-code语言（具体语言描述见《编译原理》（第3版），清华大学出版社，王生原等编著）。

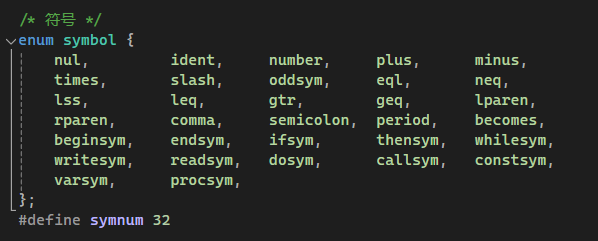
# 实验步骤

1. 编写头文件pl0.h。

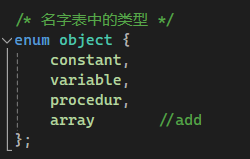
定义关键字个数，名字表容量，number的最大位数，符号的最大长度，地址上界，最大允许嵌套声明层数，最多的虚拟机代码数。



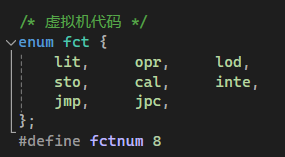
定义所有可以被识别的符号



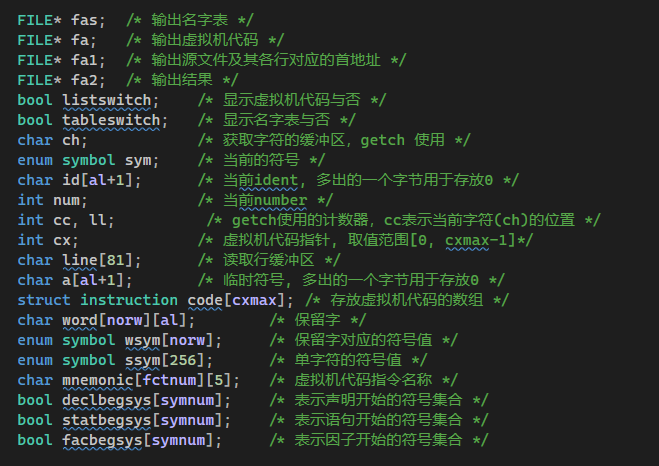
定义名字表中的类型

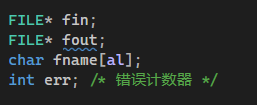


定义虚拟机中的代码

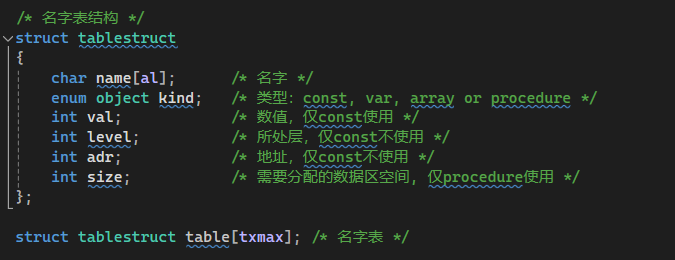


定义其他需要的全局变量

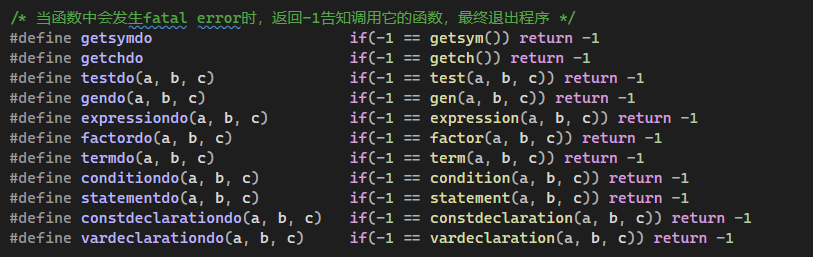




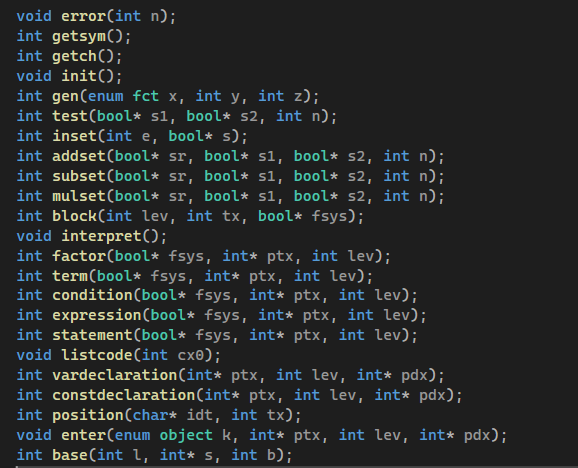
定义名字表的结构和名字表



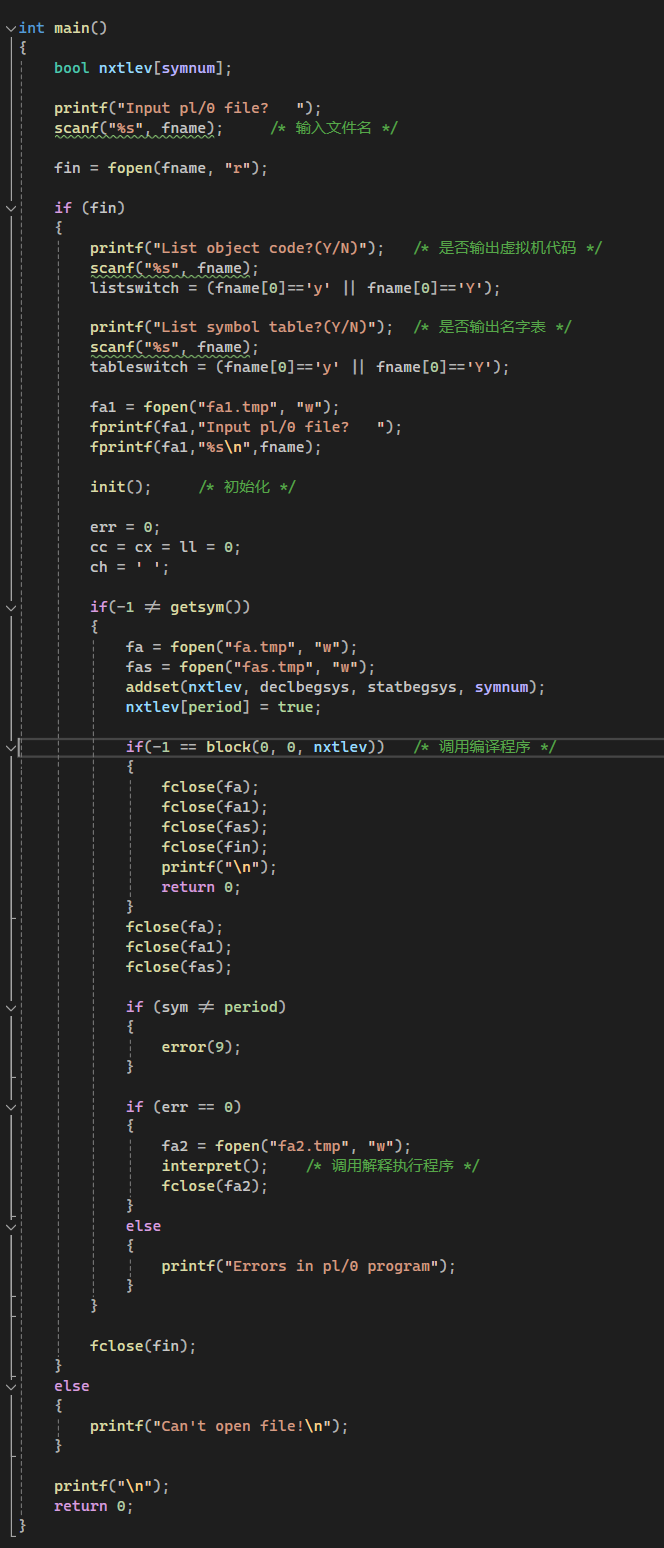
定义部分函数的异常退出



定义所有需要编写的函数接口



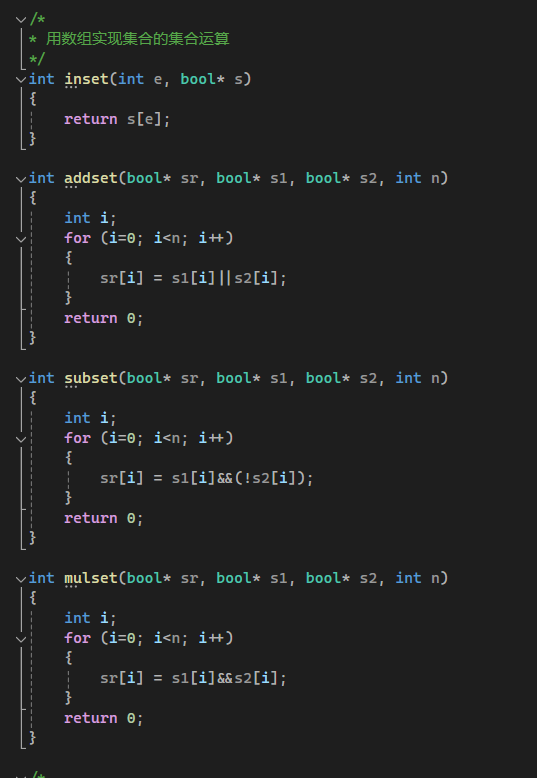
1. 编写主函数



1. 初始化符号集合、保留字和虚拟机指令集



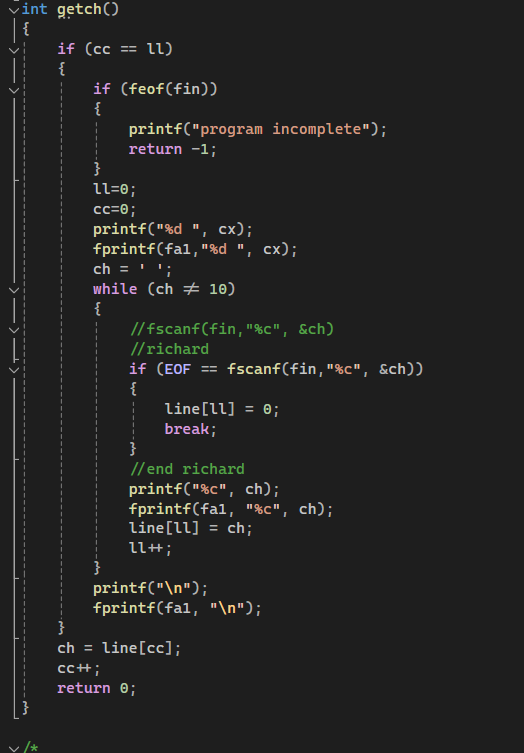
1. 编写用数组实现的集合运算函数



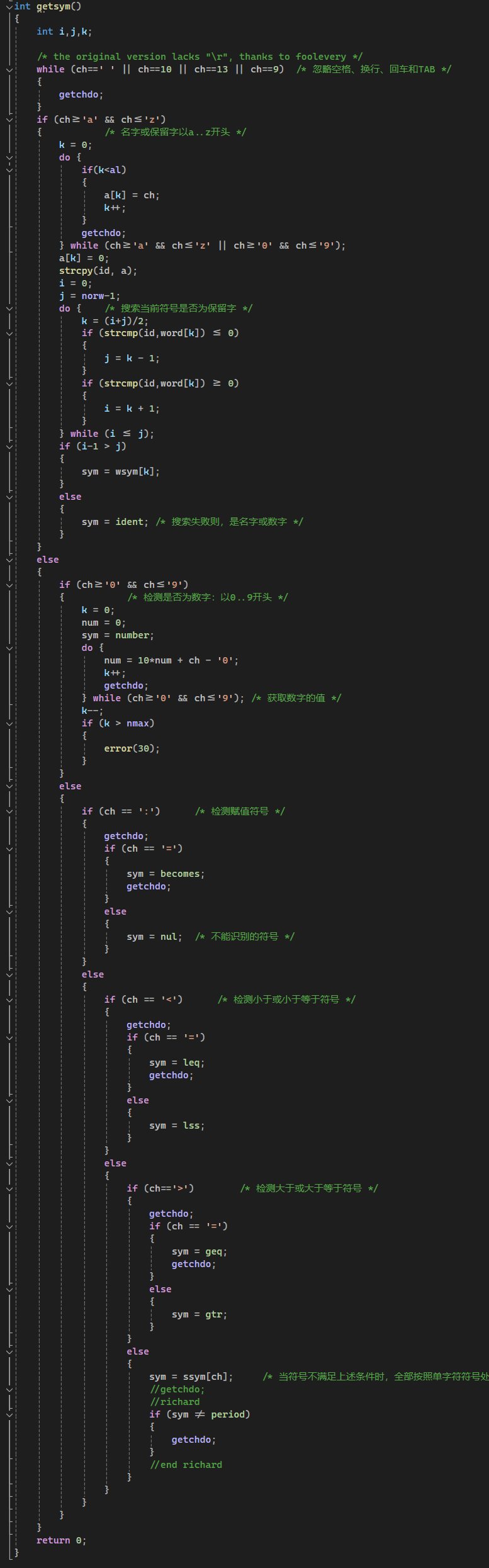
1. 编写出错处理函数



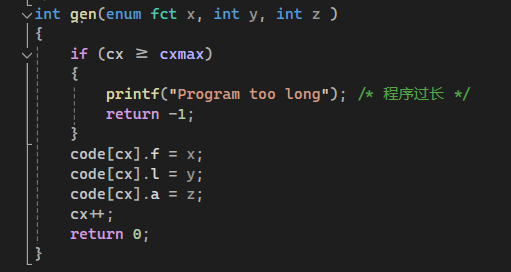
1. 编写字符读取函数，无视空格读取一个字符



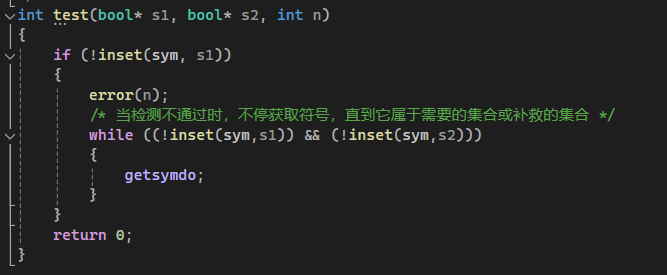
1. 编写token获取函数



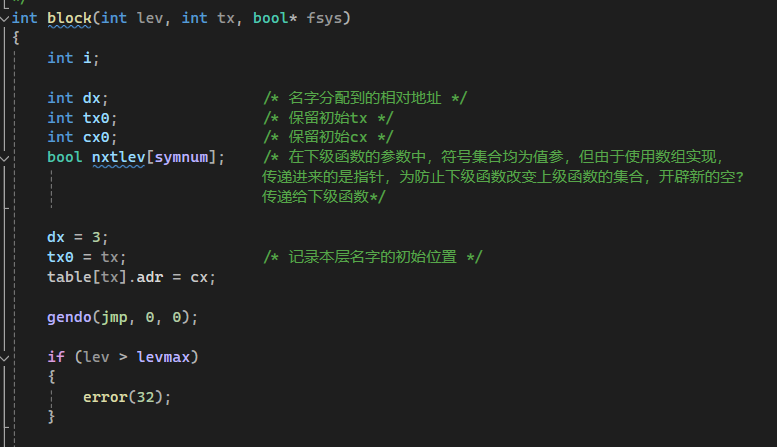
1. 编写代码生成函数



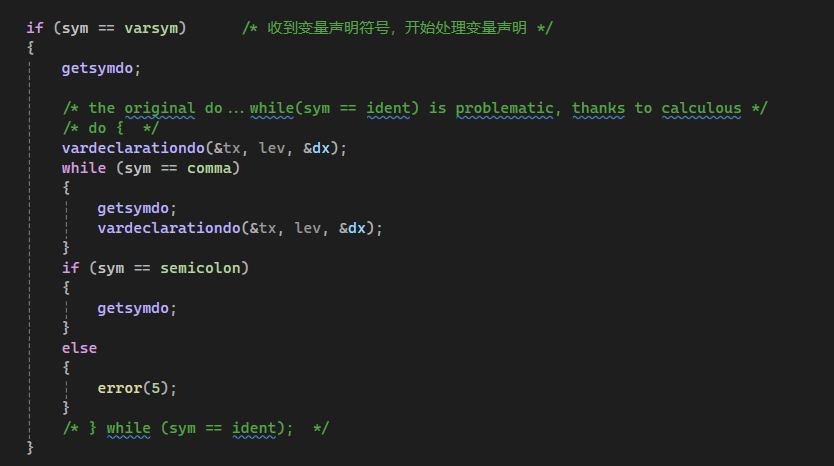
1. 编写测试函数，测试当前符号是否合法



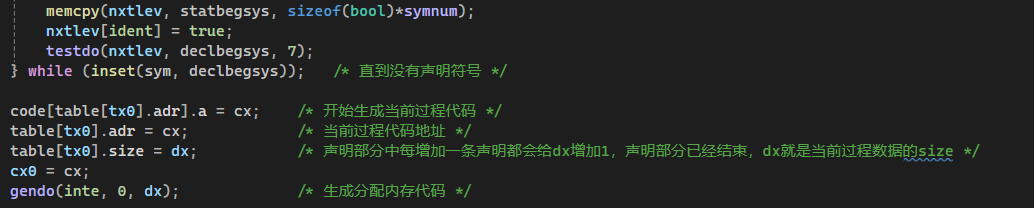
1. 编写编译主程序bolck()

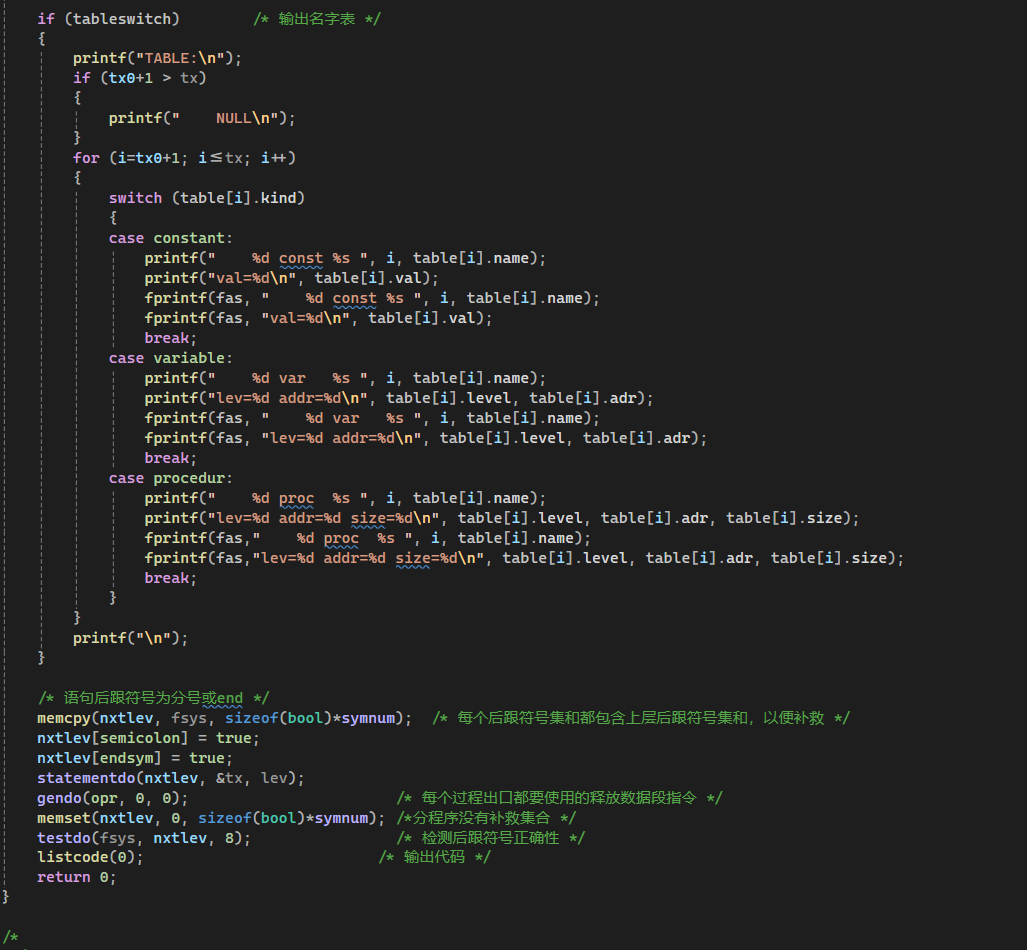




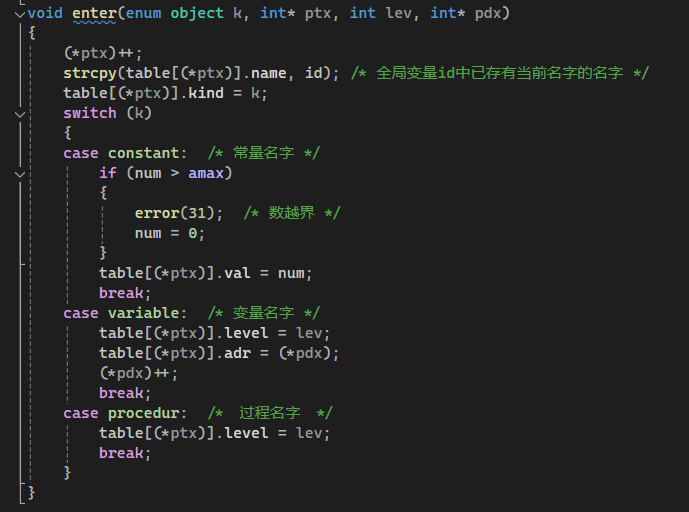




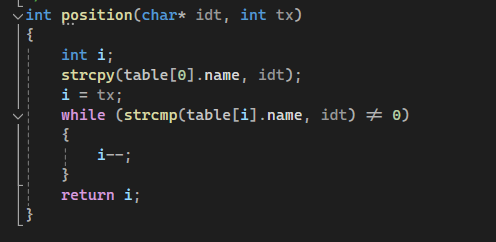




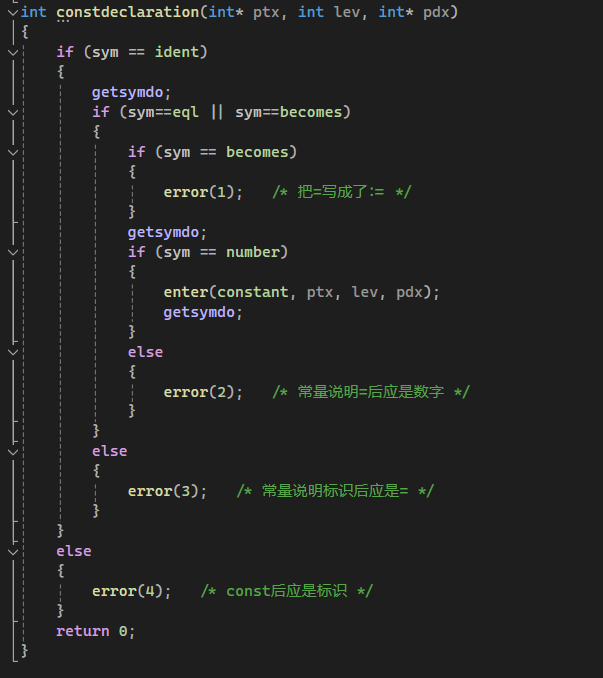
1. 编写enter()函数，在名字表中加入一项



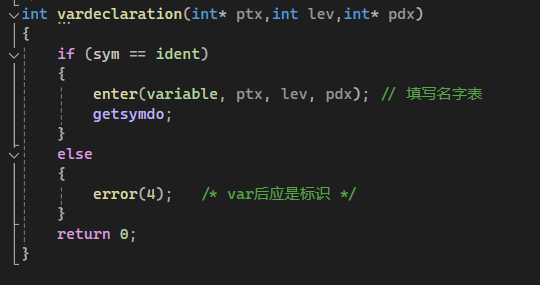
1. 编写查找函数，查找在名字表中的位置



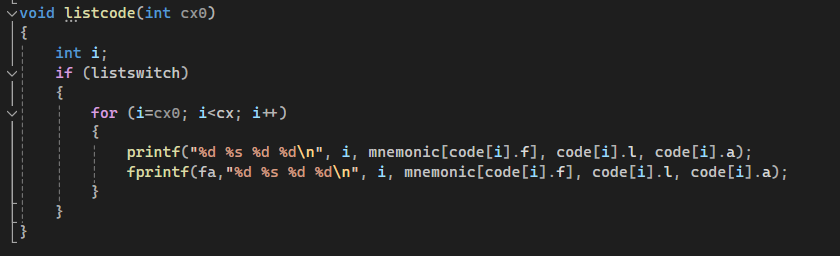
1. 编写常量声明处理函数



1. 编写变量声明处理函数



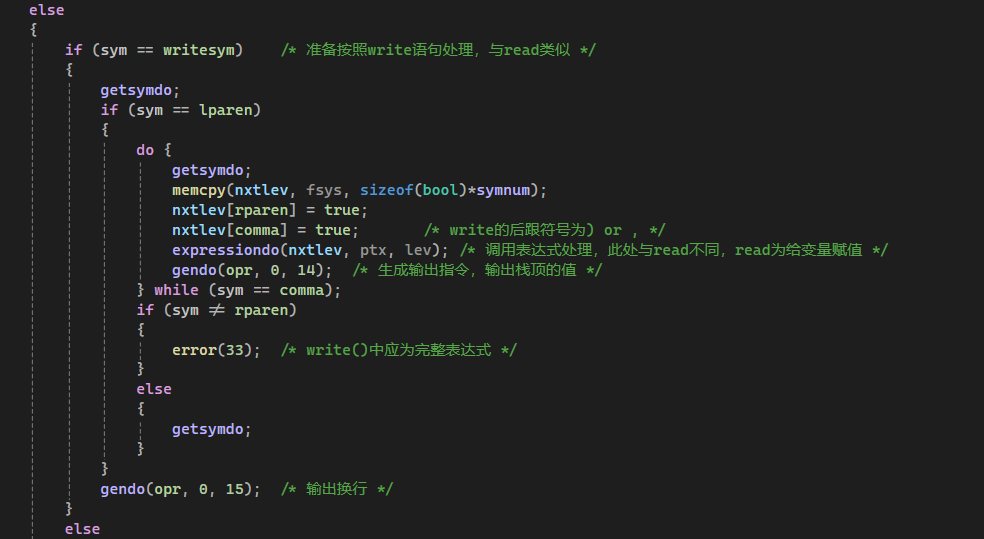
1. 编写目标代码输出程序



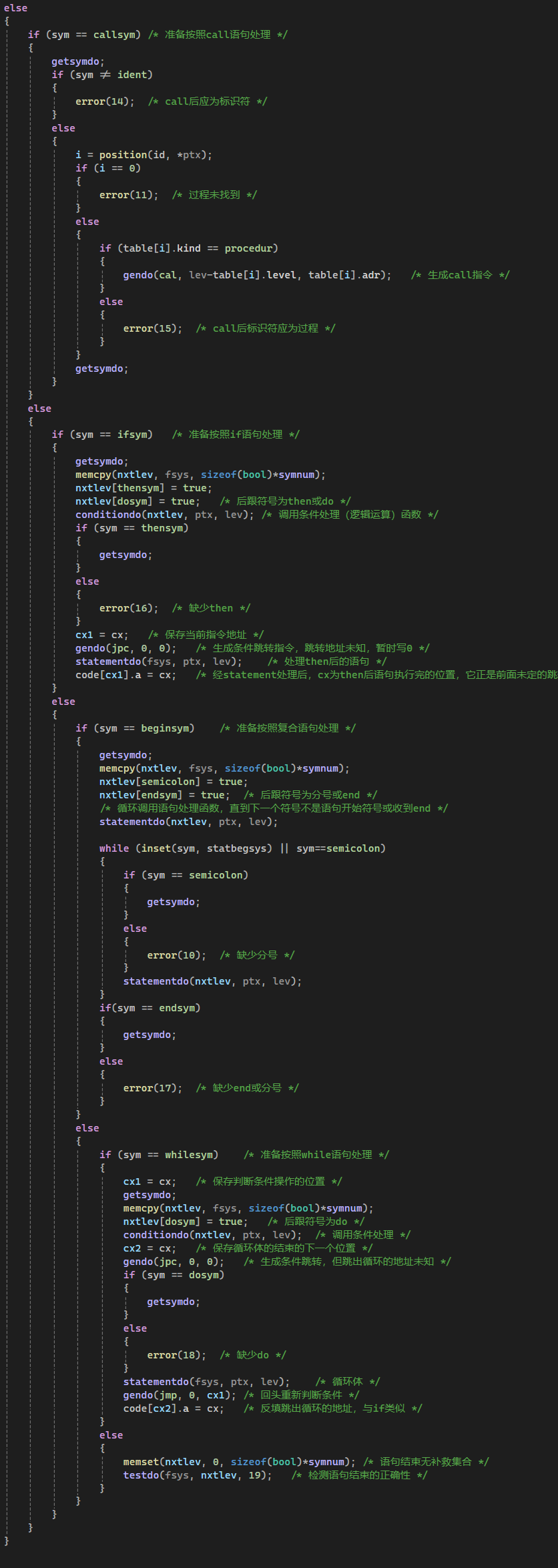
1. 编写语句处理函数







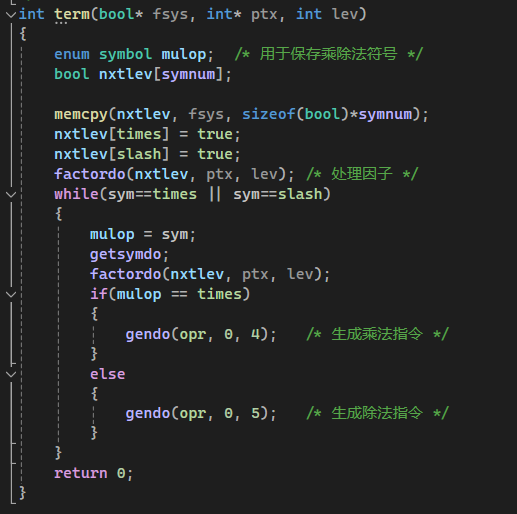




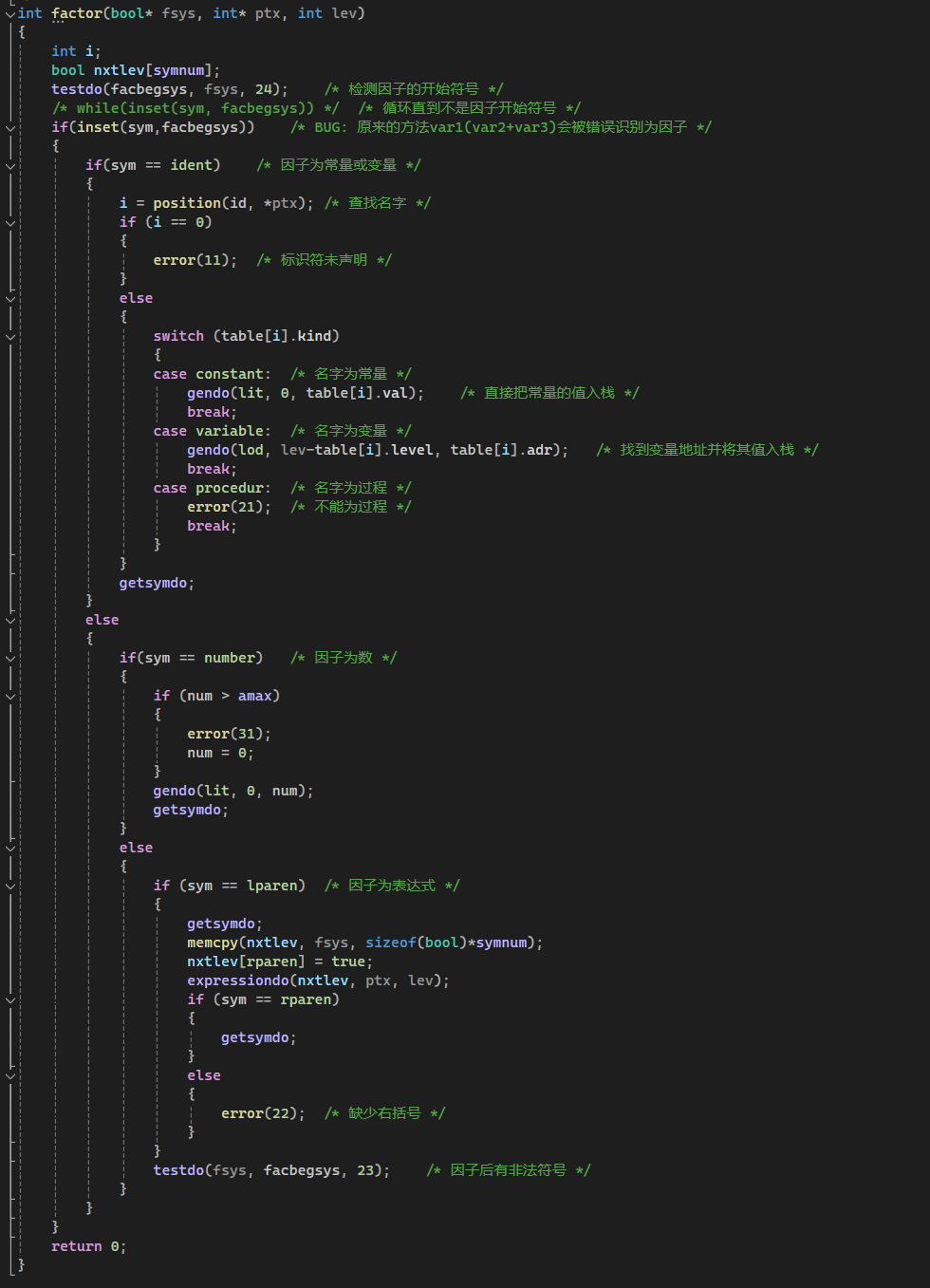
1. 编写表达式处理函数



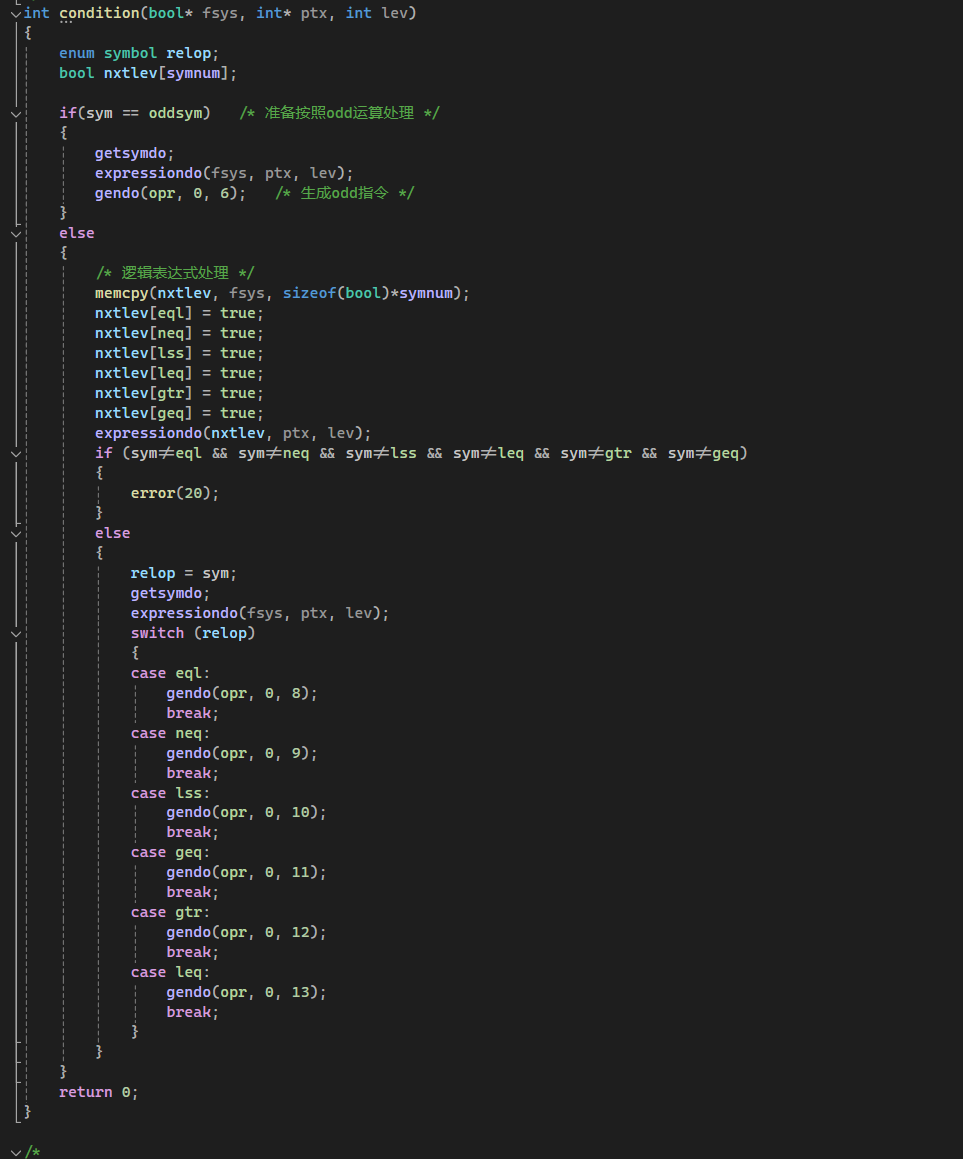
1. 编写项处理函数



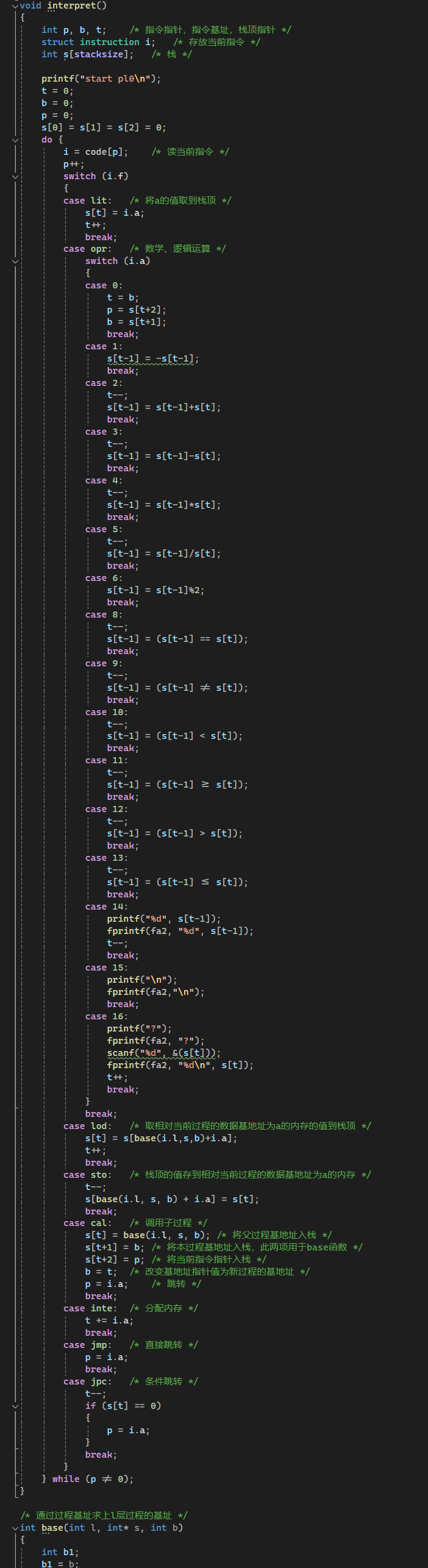
1. 编写因子处理函数



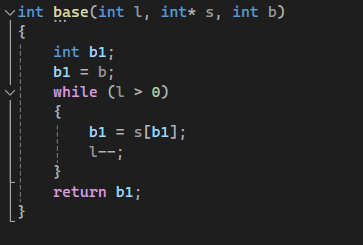
1. 编写条件处理函数



1. 编写解释程序

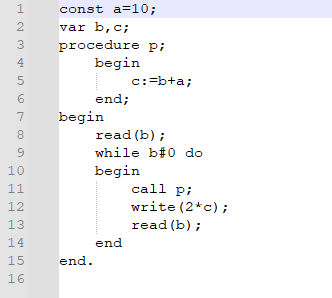


1. 编写过程基址求上层基址函数

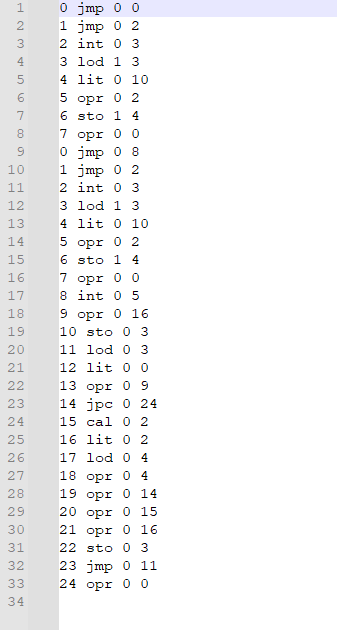


# 实验结果

输入文件：



输出虚拟机代码：



# 我的体会

在设计和实现一个PL/0编译器的过程中，我获得了许多宝贵的经验和体会。

1、深入理解编程语言原理：通过编写PL/0编译器，我深入理解了编程语言的语法结构、语义和编译原理。这包括了词法分析、语法分析、语义分析以及代码生成等方面的知识。

2、掌握了编译器设计与实现的基本流程：在编写编译器的过程中，我逐步学习了编译器的基本流程，包括词法分析器的设计、语法分析器的设计、语义分析器的设计以及代码生成器的设计等。

3、提高了代码设计和优化能力：编写编译器需要考虑到代码的效率和可维护性，我学会了如何设计清晰、高效的代码结构，并对代码进行优化以提高编译器的性能。

4、增强了问题解决能力：在编写编译器的过程中，我遇到了许多问题和挑战，如语法歧义、性能优化等。通过解决这些问题，我提高了自己的问题解决能力和技巧。

5、加深了对计算机体系结构的理解：编写编译器需要深入了解计算机体系结构和指令集，尤其是对于P-code语言的生成，我学会了如何将高级语言转换成底层的指令序列。

总的来说，设计和实现一个PL/0编译器是一次极具挑战性和收获的经历，它不仅提升了我的编程能力和理论水平，也让我更深入地理解了计算机科学的核心原理。