

**编译原理实验报告**

题 目 中间代码生成

学 院 计算学部

专 业 软件工程

学 号 120L022109

学 生 李世轩

任 课 教 师 陈殷

哈尔滨工业大学计算机科学与技术学院

2023.5

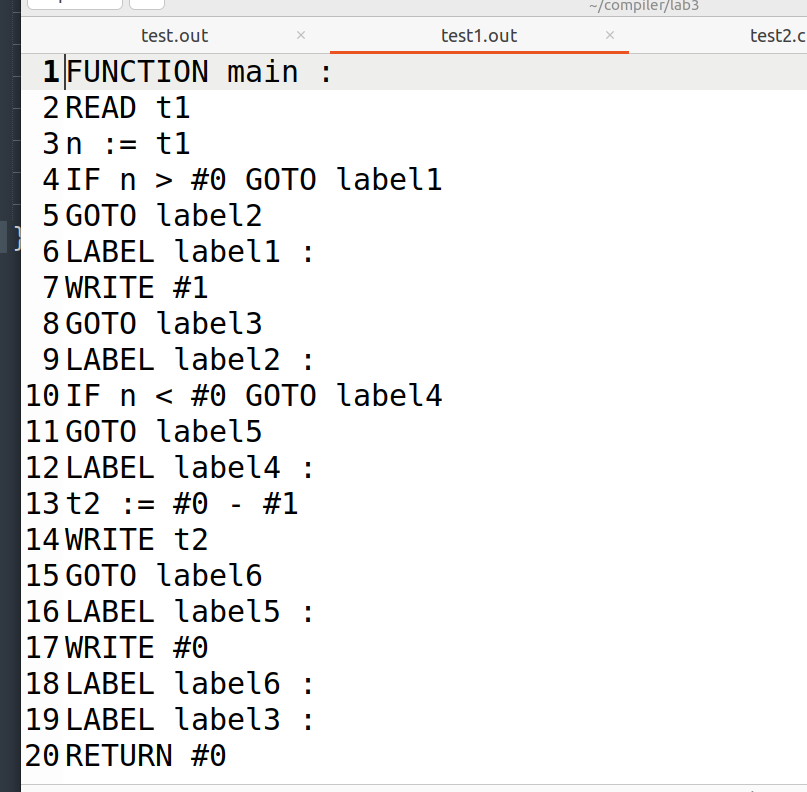
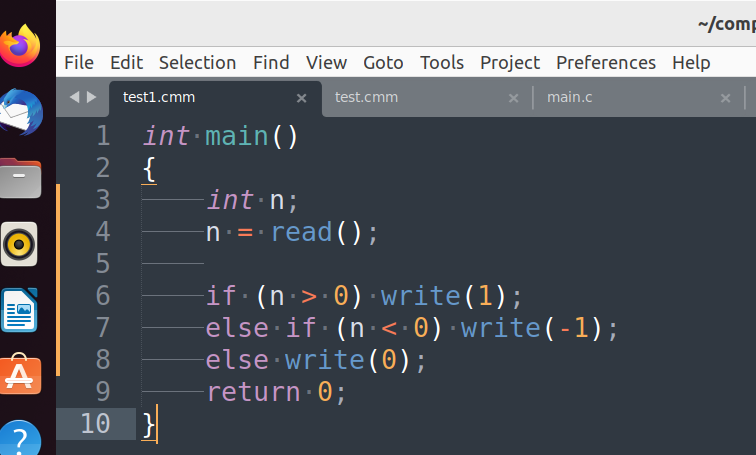
a）程序实现了哪些功能？简要说明如何实现这些功能。清晰的说明有

助于助教对你的程序所实现的功能进行合理的测试。

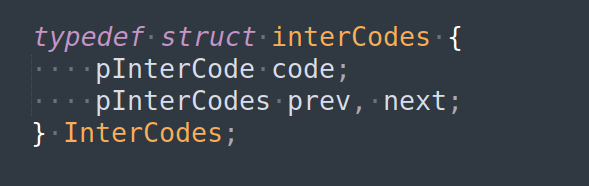
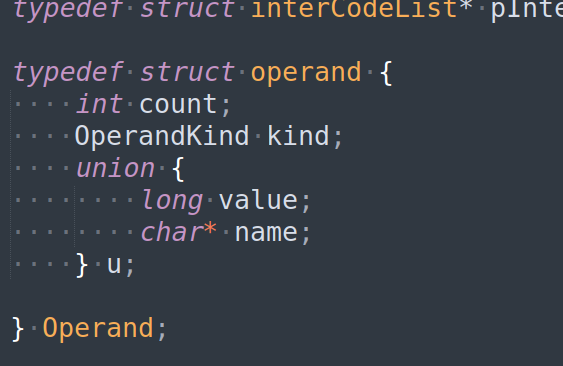
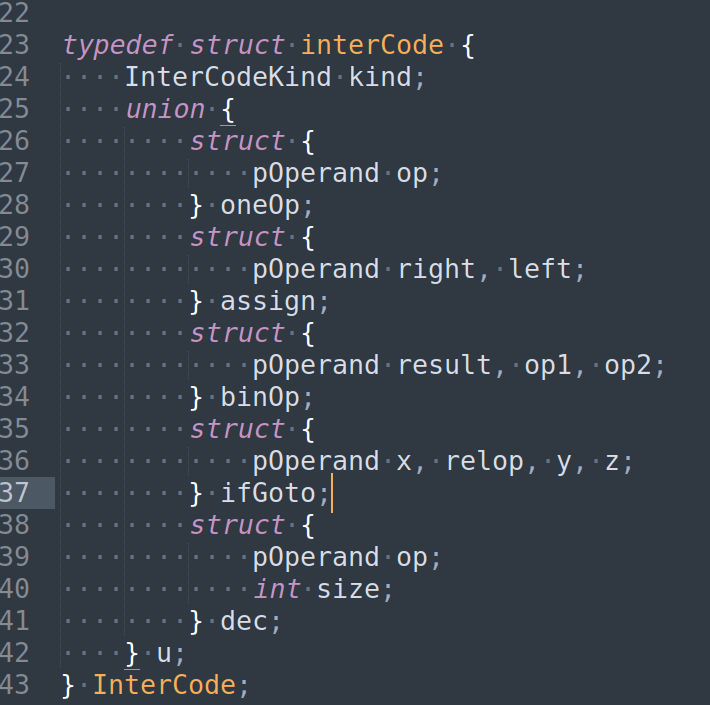
程序的输人是一个包含C--源代码的文本文件，程序需要能够接收一个输入文件名和一个输出文件名作为参数。假设程序名为parser、输入文件名为test.cmm、 输出文件名为test.out，程序和输人文件都位于当前目录下，那么在Linux命令行下运行./parser ./test.cmm ./test.out即可将输出结果写入当前目录下名为test.out的文件中。

实验三要求程序将运行结果输出到文件。输出文件要求每行一条中间代码， 每条中间代码的含义如前文所述。如果输人文件包含多个函数定义，则需要通过 FUNCTION语句将这些函数隔开。FUNCTION语句和LABEL语句的格式类似， 具体例子见后面的样例。

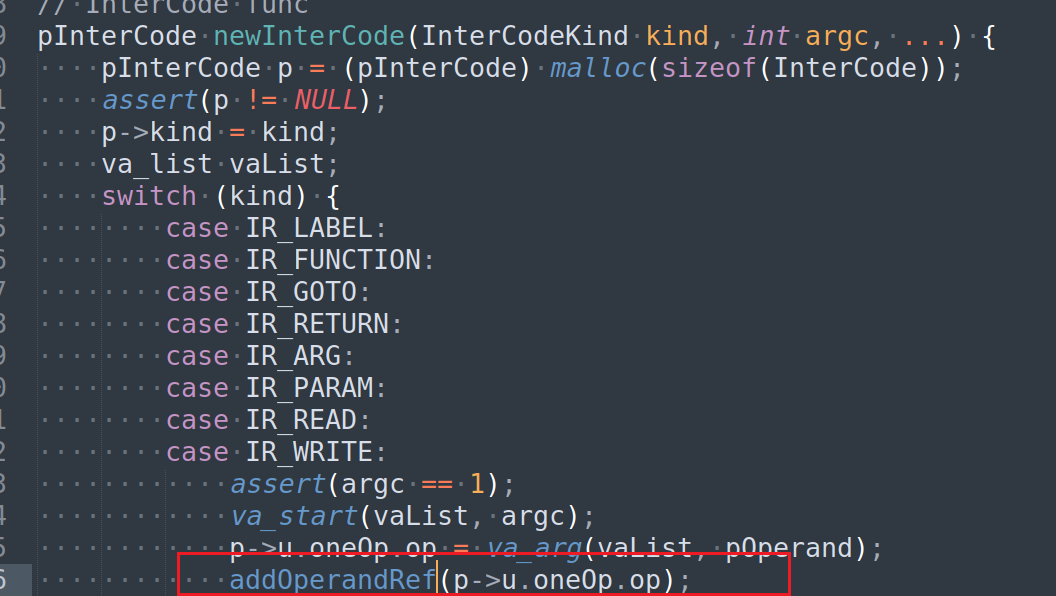
比如对于这样的输入，会有如下输出。

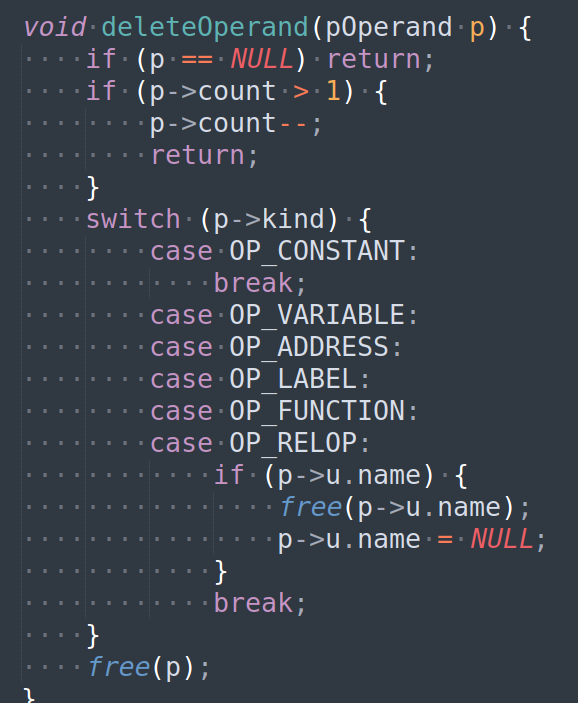


关于程序的实现，首先介绍一下定义的一些结构体。



这其中比较特殊的是operand的定义中的count，这里使用应用计数技术，避免同一块内存被多次释放。这是因为，在生成代码过程中，有很大可能会有多条中间代码指向同一个操作字，在释放动态申请的内存时，如果不使用这样的处理，就无法合理的释放内存。具体对这个域的使用方法如下，每次将一个operand放入某一个语句时，可以将这个值加一，而当释放operand时，若这个值大于一，则只要将其减一即可，只有当这个值为一，才真正释放这块内存。





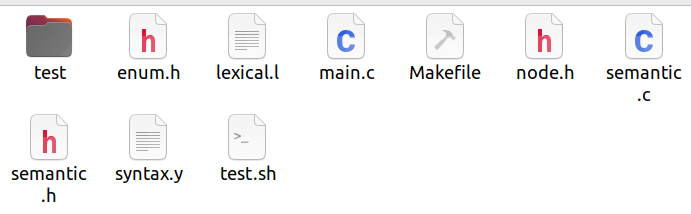
另外，程序对结构体的一些操作，结合面向对象的思想，进行了封装。



b）程序应该如何被编译？可以使用脚本、makefile或逐条输人命令进行

编译， 请详细说明应该如何编译你的程序。

程序目录如下



编译使用makefile，即切换到源程序目录下，使用make指令。