

使用实例

完成情况：

根据老师的课设完成要求，前两项任务**全部个人独立开发设计完成（支持查重）**，并全部按起评分最高的推荐方法完成程序编写和文法设计，其中：

1.任务 1 词法分析程序在识别 5 种 token 的既定任务之外，为了最大限度地贴合真实高级语言要求规则，精心设计词法分析 2 型文法，**实现了对 token 中不同常量（整型、浮点型、复数）、关键字、界符、标识符的合法性检查以及分类**，并将常量细化为**科学计数法、10 进制、8 进制、16 进制、复数、普通浮点数、且全部支持各自的合法性检查**，通过文法设计和编程结构的优化**最大限度地贴合实际编程规范**。

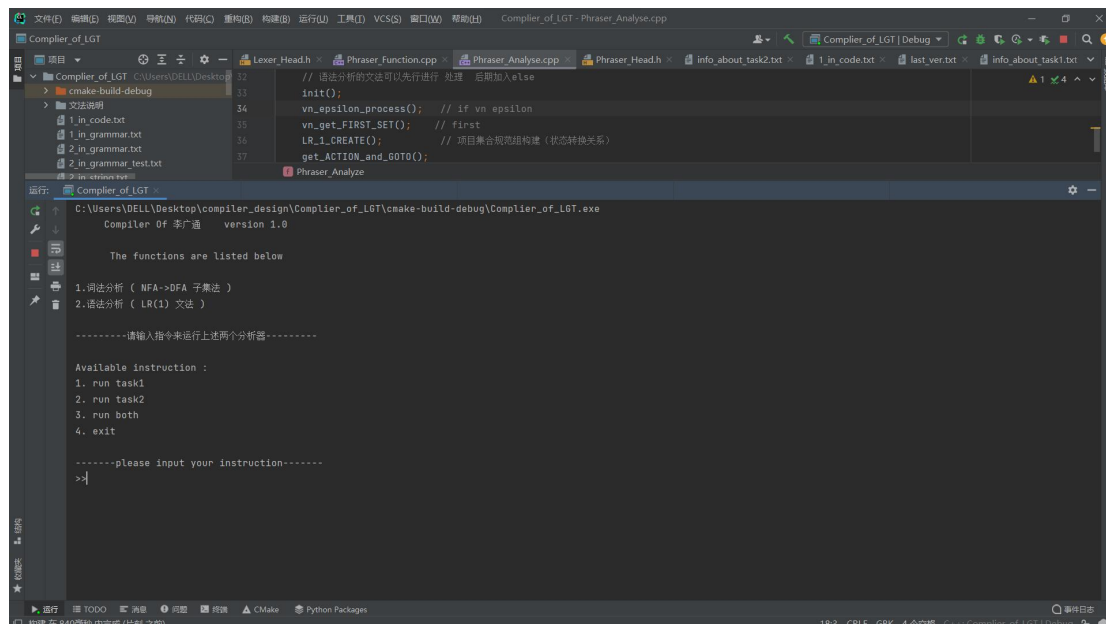
2.任务 2 文法分析选择了起评分高的 **LR (1)** 分析法，实现基础要求的基础上，还实现了**语法错误的识别标注相关行号并输出错误信息**，结合任务一显示可能的错误信息，通过语法分析的文法设计，尽可能地**包含大部分机动性编程情况**，满足可以随意多类型变量声明、赋值、循环嵌套、多头文件声明、主函数单一出口、变参个数计算等功能，对任务 1 的输入要求更加的包容，过程信息运行时**即时输出各种信息**。

整个开发过程中的**文法设计占据重要地位**，为了完成更加丰富的功能，文法改进迭代了很多版本；编程实现使用了**丰富的 STL 模板类并进行自定义重载**，利用其**高效查询以及去重**的特性，非常适合高效的分析过程，逻辑更加清晰；整个编程的代码实现没有针对特定文法（除任务 1 DFA 转化后部分），换用其他文法同样有效；程序运行时的**过程信息和提示都分别实时输出在 txt 文本和 terminal 中**，体现**程序分析的完整性（文法的 V_n 、 V_t 、DFA 终态组成、状态转移表、能否推出空、First 集、项目集规范组、项目集转化关系、ACTION-GOTO 表、LR(1)分析的过程信息）**

下面给出软件课设的使用实例和全流程效果演示

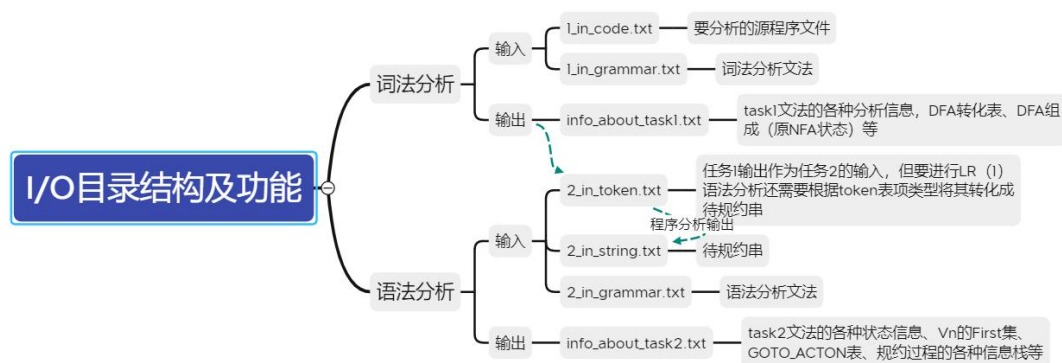
打开.exe 文件后出翔如下提示信息：

软件功能和自定义的指令，可以随时独立进行词法分析或语法分析，也可以进行连续分析，方便中间根据报错信息对源代码进行更改而不用重新关闭开启程序。



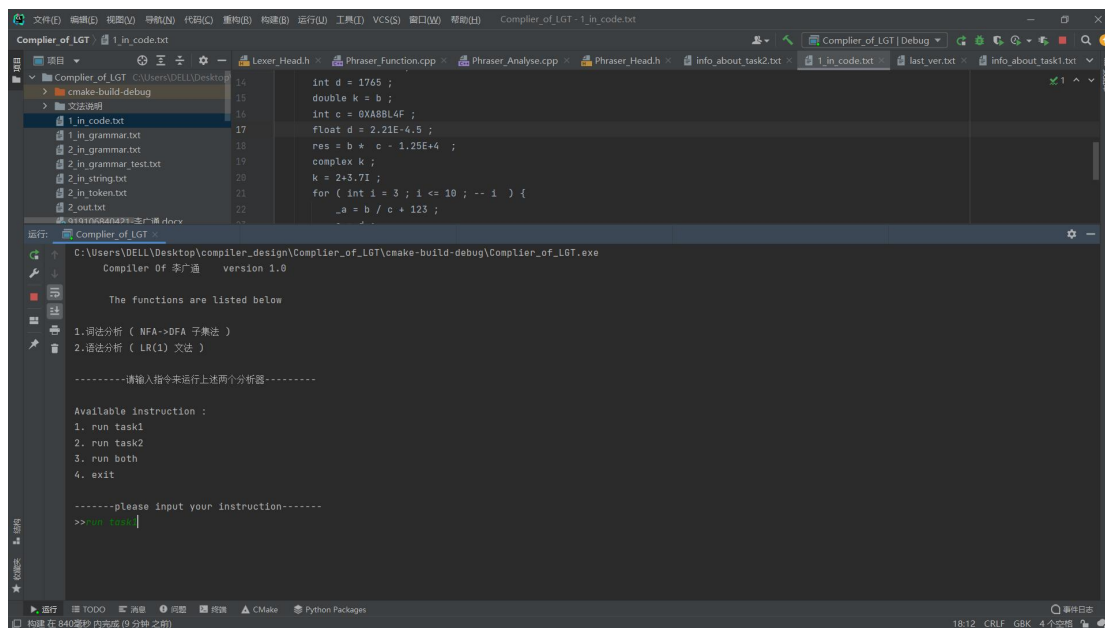
其中软件的关键输出（例如报错信息和项目集规范簇、GOTO—ACTION 表才在 terminal 中输出），过程信息（非常全面，文法分析的各个步骤和中间结果以及语法分析过程，转化矩阵等全部在 info_about_task1.txt 和 info_about_task2.txt 中），token 表也设置了专门的 2_in_token.txt 进行显示。

为方便理解输入输出的文件结构，方便老师查看，文件功能如下

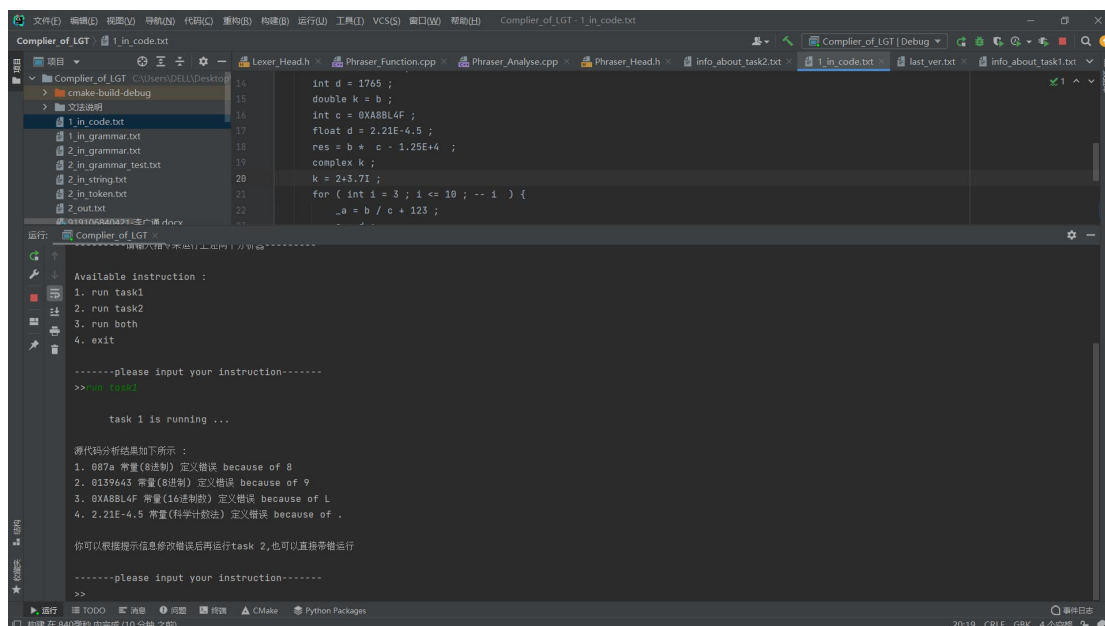


下面正式进入实例演示

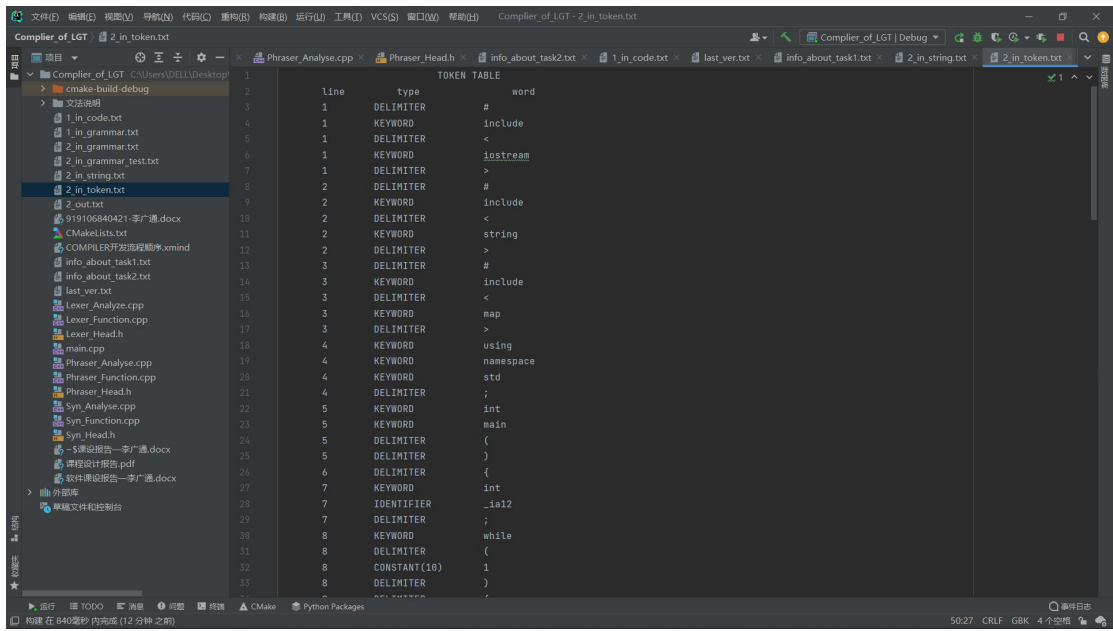
先输入 **run task1** 指令，执行词法分析器



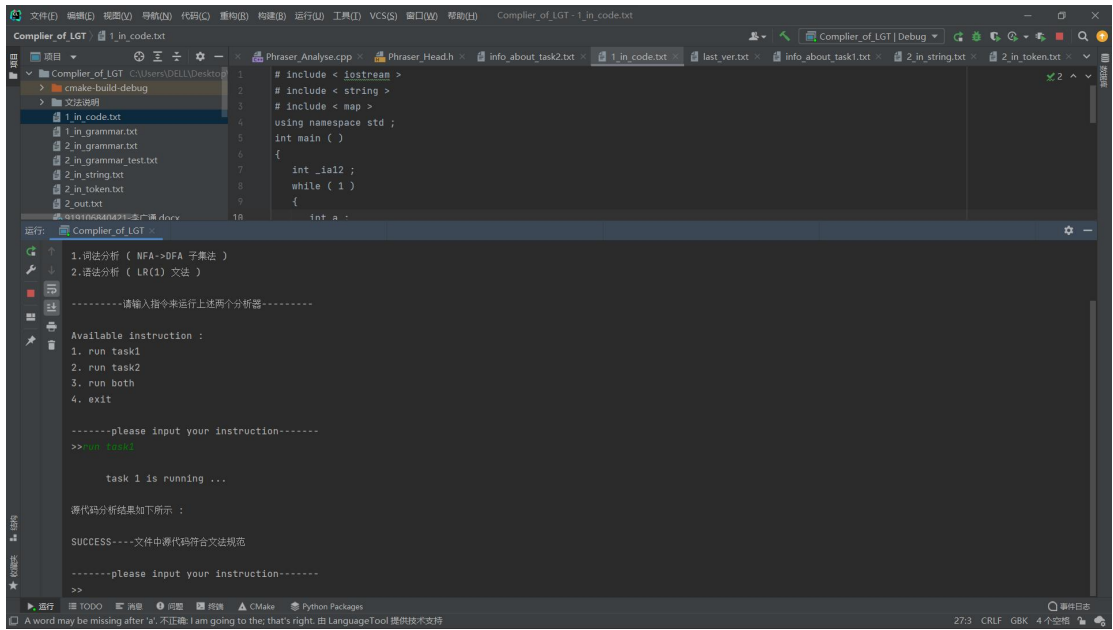
输出错误信息和详细的错误信息在控制台，可以根据提示对源程序进行修改而无需推出程序，token 文件也按格式输出。

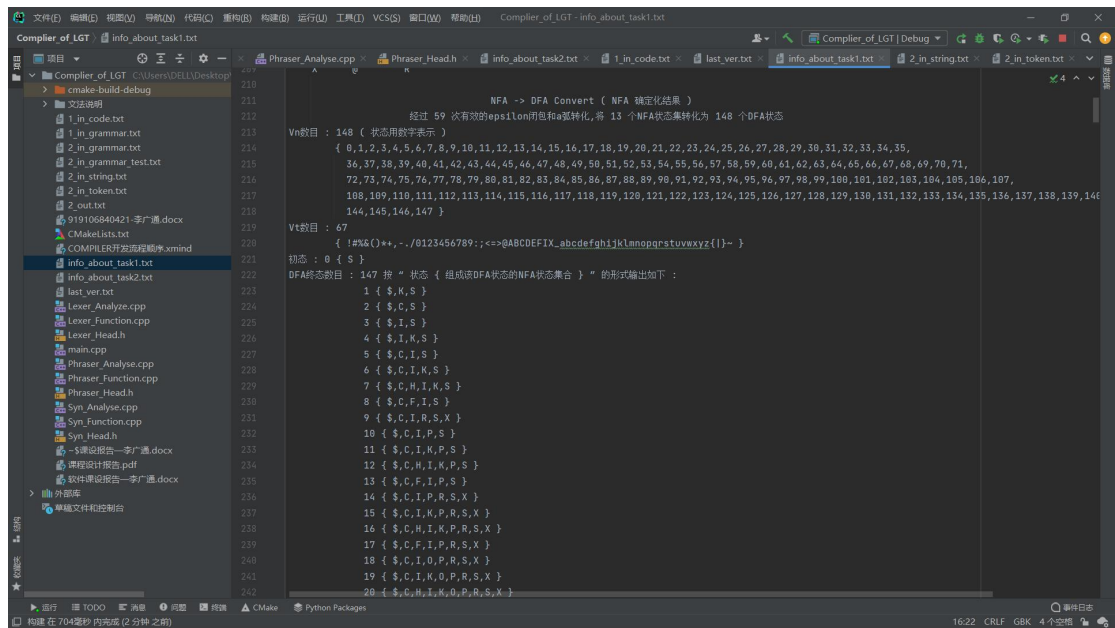


Token 表如下所示

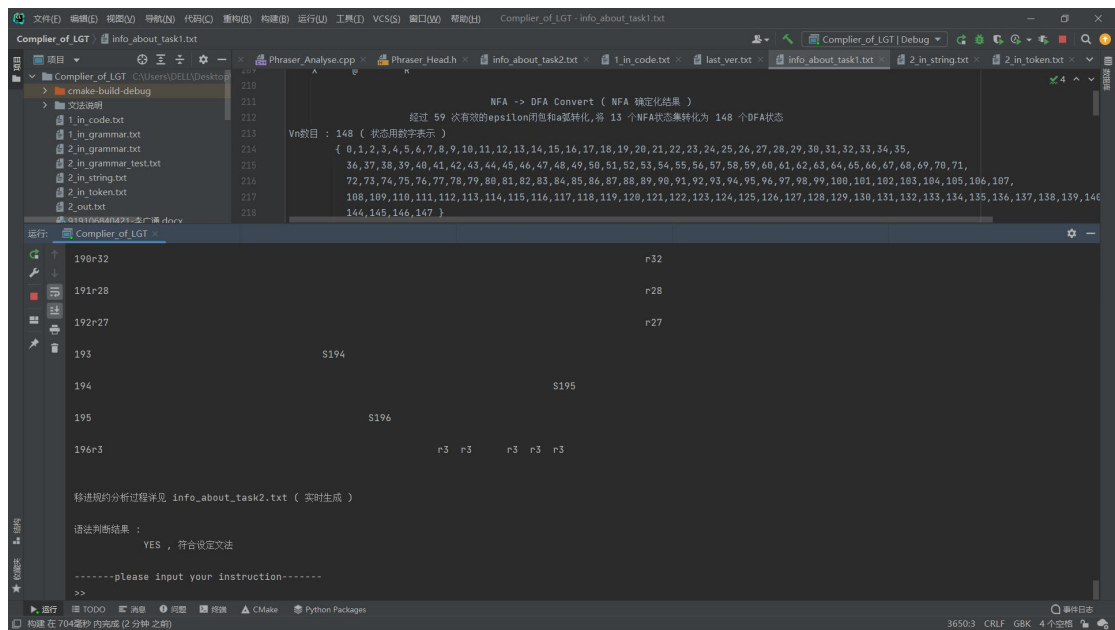


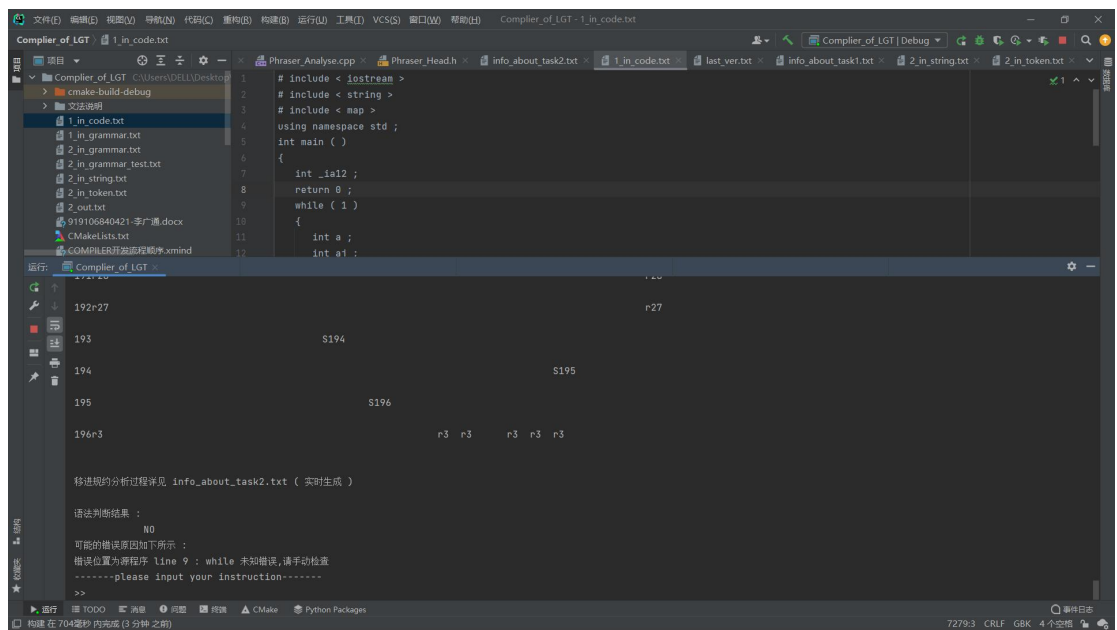
根据提示修改后，符合语法规则，info_about_task1.txt 显示各种相关信息和状态转移关系以及状态组成。





再输入 **run task2** 指令, 执行词法分析器, 输出 GOTO-ACTION 表和结果, 如果错误, 返回错误信息。





相关信息在 info_about_task2.txt 显示各种相关信息和状态转移关系以 FIRST 集合组成、项目集规范族。

