编译小组作业

实现一个**编译器/翻译器/代码高亮与提示程序**（组队完成，原则上成员不超过 4 人）。

词法分析与语法分析部分提交截止日期：北京时间 2024 年 11 月 17 日晚 23:59
完整作业提交截止日期：北京时间 2024 年 12 月 22 日晚 23:59

作业内容说明——编译器、翻译器

在该选题中，同学们需要实现一个简单的**编译器**（后端部分可以直接使用已有工具）或**翻译器**。

具体而言，给定某种语言（记为**源语言**）写的源程序作为输入程序，同学们需要进行词法、语法分析，生成与输入程序对应的输出程序（中间代码或另一种语言写的程序，输出程序的语言记为**目标语言**），并运行输出程序。

该作业中需要**实现符号表**，**在语法错误、变量未定义等情况下报错并给出修改建议**。

源语言（即输入程序的语言）可选择如下：

1. Java；
2. JavaScript；
3. C/C++；
4. C#；
5. Python；
6. 自己设计的编程语言，难度+1；
   （需提前联系助教确认，说明该语言的特点、语法并给出代码示例）
7. 其他（需提前联系助教确认）

目标语言（需与源语言不同）可选择如下：

1. Java 字节码（可在 Java Virtual Machine(JVM)上执行的.class 文件）；
   （可先由同学们的程序生成.java 文件，再由 javac 编译成.class 文件）
2. MSIL（类似于 Java 字节码，是.NET 对应的中间语言），难度+2；
   http://en.wikipedia.org/wiki/Common_Intermediate_Language
3. LLVM（LLVM 架构的中间代码，可以使用 LLVM 工具编译成机器代码），难度+2；
   （推荐同学们学习 LLVM http://llvm.org/ ）
4. JavaScript（可以直接在浏览器或控制台的 JS 解释器中执行）；
5. Python（可直接被 Python 解释执行的程序）

考虑到同学们在有限的时间内不太可能支持源语言的所有语法，本次实验中仅需同学们的编译程序支持部分语法。为方便评判，每组同学需选择 2 个测试用的源语言的程序（自行用源语言实现），基本要求是实现所选程序的编译/翻译即可。

测试程序可选择如下：

1. 回文检测： 输入一个字符串 s，判断 s 是否为回文， 输出'True'/'False'。
   回文字符串是按位置中心对称的字符串，如 'a', 'aba', 'abbcdcbba' 等；

2．排序：输入若干个整数，将它们按照从小到大的顺序排序后重新输出。
例如：输入'5,8,4,9'，输出'4,5,8,9'；
3．KMP 字符串匹配：输入一个字符串 s 和一个模板串 t，在 s 中匹配 t，将所有匹配上的子串的起始位置输出；若 s 中无 t，则输出'False'。
例如：s='abcdefgabdef'，t='ab'，输出'0,7'；
4．四则运算计算：输入一个字符串 s（仅含数字 0-9 和符号+-*/()），输出该表达式的值；可能会使用到逆波兰表达式，需实现栈结构。
例如：s='1+(-5-22)*4/(2+1)'，输出'-35'，难度+1。

注意：**鼓励同学们自行提供更多测试程序，以表明所完成的编译器/翻译器能实现的不同于上述四个测试用例用到的语法，甚至是所选源语言原本不支持的新语法。**视实现情况至多可获得难度+2。

作业内容说明——代码高亮与提示程序

给定某种语言（记为源语言）写的源程序作为输入程序，同学们需要进行词法、语法分析，实现**代码高亮功能**与**代码提示功能**。输入源语言写成的代码文件（可能为一个或多个文件，如果支持引用其他文件中的类和函数可以获得难度+1），需要实现：

1．**代码高亮功能**：程序应当至少能够识别出注释、变量名或函数名、包名或类名、字符串常量、数值型常量、运算符、关键字，并且用不同的颜色表示出结果；
2．**代码提示功能**：基于已经输入的代码文件，输入一句不完整的代码，能够实现：
   a）**代码补全**：输入内容不全的时候，能提示出可能要使用的变量名/函数名/类名（如：使用 python 语言情况下，输入'prin'能提示'print'）；
   b）**参数提示**：输入一个函数名（要求至少能支持代码文件中定义的函数、代码文件中定义的类内的函数。如果能支持引入的外部包的函数，可以获得难度+2），能提示出该函数的参数。

实现上述功能时，可以选择：

1．直接在命令行中打印出结果，如：
   a）代码高亮功能中用 HTML 标签表示代码高亮的颜色；
   b）代码提示功能中将提示内容在命令行中输出；
2．制作成 VS Code 等编程工具的插件（制作成插件形式可以获得难度+2）

该作业中需要**实现符号表**，**在存在语法错误、变量未定义等情况下提示错误信息并尽量跳过错误部分**。

源语言（即输入程序的语言）可选择如下：

1．Java；
2．C++；
3．C#；
4．Python；
5．自己设计的面向对象编程的语言，难度+1；
（需提前联系助教确认，说明该语言的特点、语法并给出代码示例）
6．其他面向对象编程的语言（需提前联系助教确认）

选择该题目的同学需自行编写测试代码文件，其中需要包含：

1．引入其他包的语句（如 C++中的#include 和 Python 中的 import）；
2．类的定义（定义的类中需包括函数）及对应变量的初始化；
3．函数定义及其调用；
4．类变量对应的成员函数的调用；
5．字符串常量与数值型常量；
6．算术运算符和逻辑运算符；
7．条件判断语句与循环语句

注意事项

1. 翻译器的选题中不可以使用相同的源语言和目标语言，例如 Python->Python 或 Java->Java 字节码。
2. 本次实验程序中务必使用正规的词法、语法分析来完成作业，**不能简单地使用全局字符串匹配来代替词法分析或语法分析**，否则将按 0 分处理。切记！
3. 本次实验可使用 ANTLR、Lex/YACC(包括 PLY(Python Lex-Yacc))等词法、语法分析工具简化词法、语法分析过程，但不使用这些工具可以获得难度+3。
4. 本次实验中源语言的文法规则需要自己总结，如果参考网上已有的文法规则需要在文档中注明并给出链接。
5. 请注意，编译器在进行处理时会有预处理步骤，如 C 语言进行处理时会在预处理阶段将 include 语句和 define 语句进行处理。简单起见，本次作业中，可以将系统库文件（如：C 的 stdio.h、python 的 math）的内容改写为 用到的函数的声明语句，如：将 stdio.h 文件内容替换为 printf 和 scanf 两个函数的声明语句；
6. 一些库函数（如：scanf 和 printf 等输入输出函数、数学运算函数）处理时，可以直接当成一个函数节点，词法分析时标记为一个内置函数，语法分析时当作一个函数类型的节点，翻译器处理时可以转换成目标语言对应的函数的写法；
7. 源语言的词法语法定义（如 antlr4 中的 g4 文件）需要自己完成；
8. 本文档中标注有"难度+x"的项，代表选择该项有对应的加分，但所有加分不会使分数超过编译小组作业部分总分。

选题说明

1．编译小组作业需在腾讯文档中进行组队并选题，请写明小组成员**姓名及学号**，选择选题类型（**"编译器"、"翻译器"**或**"代码高亮与提示程序"**），组队链接为：https://docs.qq.com/sheet/DWmJqRktocUVnd1JJ?tab=BB08J2 。

2．选题截止时间为：北京时间 10 月 13 日晚 23:59。

3．代码工作量的评估与小组人数成正比，希望每个同学在小组中都有足够多的贡献。

4．编译小组作业将会安排**集中展示**（时间与形式另行通知），届时每组的作业将由其它小组进行评价打分，并作为给分的重要参考。

作业提交

本次作业分两个阶段提交，分别为：

1．提交词法分析部分和语法分析部分（分数占比 30%）。
   - 词法分析部分，提交的程序需要对输入的一段代码进行处理，输出 token 流（包括每个 token 的类型、内容以及一些必要的属性，列表形式即可）；
     - 对于 Python 语言，可以将每行前的缩进信息当成一个 token，缩进长度可以看作是该 token 的一个属性；
   - 语法分析部分，提交的程序需要对输入的一段代码进行处理，输出语法分析树（以 JSON 格式输出或者 YAML 格式输出，而**不是图片等其他格式**）；

2．提交完整作业（该部分及展示表现分数占比 70%），需要在词法分析和语法分析的基础上，完成完整的编译器/翻译器/代码高亮与提示程序。

每次作业提交时，应当提交一个压缩包，命名为组员 1 姓名_组员 2 姓名_组员 3 姓名_编译小组作业.rar/zip，其中包含：

1、源代码文件，置于 src 文件夹内。

2、可执行文件（可以是 exe 程序、python 脚本等可直接执行的程序）及**测试用的源语言写的源代码**，置于 exe 文件夹内，并附上说明程序运行方式的 readme.txt 和程序输出结果。

   a) 如果是直接运行 src 中的文件，则无需将 src 中文件再次拷贝，只要在 exe 文件夹中的 readme.txt 文件内说明运行方式即可。

3、说明文档，突出难点和创新点，并写明小组分工。可以添加一些程序运行时的截图用以辅助说明。整体篇幅不宜过长，建议在 4 页以内。

每个小组有一位同学提交压缩包即可。

提示

1、 如果使用 antlr,可以阅读官方文档中对于 Listener 和 Visitor 的使用说明，可以在词法、语法分析和生成目标代码的时候用到他们。一些简要介绍可参考：https://github.com/fengh16/THSS_Compiler/blob/master/编译小组作业_ANTLR4.md 。如果使用 antlr，**不能直接用 antlr 命令行**输出 token 序列作为词法分析结果。

2、 在词法分析部分，需要**从左向右扫描输入的字符流**，可以使用正则表达式匹配剩余输入串（正则表达式对应于自动机，但**不能直接全局正则匹配关键词**）；

3、 对于编译器和翻译器的选题，在生成目标代码时，可以对应每个类型的非终结符分别写一个函数，将该非终结符的代码生成出来，如：
   - 生成 python 的条件判断语句时，if 节点下面，可能是有一个条件节点和一个内容节点；
   - 对于这个条件判断，生成的代码可以是：
     - 'if' + 空格 + 条件节点的代码 + 冒号 + 缩进函数(内容节点代码)
   - 其中，“缩进函数”是一个给每一行代码增加缩进的函数。

展示说明

编译小组作业展示的 PPT 要**突出自己小组作业的特色和难点，说明难度加分及对应的项目**，并**附以测试代码的演示**（可以使用录屏），展示时间**严格控制在 4 分钟以内**。请注意在网络学堂上对应作业窗口中及时提交 PPT 文件。

除此之外，同学们可以自行确定 PPT 具体包含哪几部分，例如：选题、开发环境（包括使用的编程语言、是否使用 antlr 等工具）、实现的功能（着重说明作业的特色）、创新点（如有）、难点及解决方法、小组分工、演示。