

大作业第一部分 词法分析

本次作业时大作业的第一部分，即开发一个词法分析器。分组完成。每组同学自行分工合作完成作业，在 deadline 之前，每组由一人将完成的系统提交到系统的相应 folder。

目标

练习使用 Lex 或者手工编写 C- (C 语言子集) 语言的词法分析器，理解词法分析器的工作原理，熟练掌握高级语言的单词符号的正规式表示，词法分析器的工作流程并编写源代码，识别出单词的二元属性，填写符号表，同时编写简单的词法错误处理程序。

软件需求

1. 完成 C- 语言的词法分析器，词法分析器的输入为 C- 语言源代码，输出识别出单词的二元属性，填写符号表，包括关键字，标识符，界符，算符，常数，其他六大类型；其中，
关键字包括 while break continue for if else float int void return main
算符包括 + - * / = >= == <=
界符包括 () { } ;
标识符，常数的定义与 C 语言相同
注意：C- 语言包括且仅包括上面限定的关键字，算符和界符，其他 C 中的关键字，界符，算符不属于 C- 语言。即若 C- 程序中出现 printf，不会被识别成关键字，其也不具有 C 中的特殊含义。
2. 可以选择基于 Lex/ANTLR 自动生成词法分析器，或者自行设计并实现词法分析器；
3. 实现语言：C/Java/Python；

提交文件要求：

1. 基于 Lex 实现的词法分析器，需要提交
 - a. 词法分析器可识别语言的 BNF 范式 (pdf 文档)；
 - b. lex.l 文件；
 - c. lex.yy.c 文件；
 - d. 最终生成的词法分析器；
 - e. 使用说明文件(readme file) (pdf 文档)；
2. 自动实现的词法分析器需要提交
 - a. 词法分析器的完整设计 (pdf 文档)；
 - b. 识别语言的 BNF 范式 (pdf 文档)；
 - c. 词法分析器的全部源代码以及可执行文件；
 - d. 使用说明文件(readme file) (pdf 文档)。

提交方式：

✓ 本科教学系统→编译原理→作业管理→大作业_词法分析

- ✓ 每组全部需要提交的文件放入一个文件夹，压缩 (.zip, .rar)，并以组名命名

压缩包，例如“第一组.zip”

截止日期(Deadline)

2018 年 11 月 6 日 11: 59pm (GMT+08)

测试用例样本

```
void main() {  
    int i=2;  
    int j=4;  
    int result=0;  
    while(i <=j) {  
        result=result+j;  
        i++;  
    }  
    printf();           //注意，在 C++ 中，这不是一个打印操作  
    return result;  
}
```

注意事项

本次作业会进行抄袭检测，对检测出抄袭的作业，予以 0 分处理。

参考文献

[1] The Lex and Yacc Page <http://dinosaur.compilertools.net/#overview>