# L1 词法分析 实验报告

181250015 陈彦泽

## 1 实现功能

- **输入一个文件，将文件读取后进行词法分析**
- **如果有未定义的词素，进行报错**

  `Error type A at Line [行号]: Mysterious character "[词素]".`
- **否则输出Token信息**

  `Token [词素] at Line [行号].`
  - 识别INT类型，包括十进制、八进制、十六进制
  - 识别FLOAT类型，包括浮点数、科学计数法浮点数
  - 识别ID
  - 识别保留字

## 2 实现方法

使用flex，主要的任务就是为：

1. 定义各个词素定义正则表达式
2. 对于行号的输出，主要使用的flex自带的`%option yylineno`

以下主要记录实验中碰到的一些难点和踩过的坑

### 2.1 优先匹配原则

根据匹配的优先原则，以下有一些必须遵守的顺序：

- 先匹配RELOP后匹配ASSIGNOP，否则无法匹配到`==`
- 先匹配保留字后匹配ID，否则无法匹配到保留字
- 先匹配注释后匹配DIV，否则`//`类型的注释无法被匹配
- 先匹配浮点数，再匹配十六进制、八进制数，最后匹配十进制数(解释见下文)

### 2.2 Token定义的细节

**十进制整数**

- 0或以[1-9]开头的数字串，不存在0开头的非0数字串

**八进制整数**

- 必须以0开头，所以匹配顺序要在十进制之前，否则例如07会被识别为INT(0)、INT(7)

**十六进制整数**

- 必须以0x、0X开头，所以匹配顺序要在十进制之前，否则例如0x1会被识别为INT(0)、ID(x)、INT(1)

**浮点数**

根据小数点的位置，可以分为三种类型

- 小数点在最前面和最后面，如`.5`、`8.`，后面必须跟科学计数法，即`.5E3`、`8.E-4`等

  `(\.{digit}+(E|e)(\+|-)?{digit}+)|({digit}+\.(E|e)(\+|-)?{digit}+)`
- 小数点在中间，如`3.5`，科学计数法可有可无

  `({digit}+\.{digit}+((E|e)(\+|-)?{digit}+)?)`

## 2.3 转换

- 将不同进制的数转为十进制整数输出
  - 自己写了`hexToInt`和`octToInt`的方法进行转换
- 将科学计数法转换成浮点数输出
  - 使用c自带的函数`atof(yytext)`进行转换即可

## 2.4 错误输出

如果有未定义的词素，就直接报错了，不输出TOKEN了

所以在识别的时候可以用yyout先把中间结果存到文件中，读取结束后再根据error标记位选择是否输出TOKEN